MTR110358

MITRE TECHNICAL REPORT

MITRE

KML Best Practices for Interoperability

Jason Mathews
September 2011

Abstract

Keyhole Markup Language (KML), the data format used by Google Earth™ to annotate and view geospatial data on maps, has become an international Open Geospatial Consortium (OGC) standard and has been incorporated into numerous commercial and government systems. KML is no longer viewed as a vendor-proprietary data format but rather as an open data standard for a growing community.

The increase in applications consuming and producing KML requires that best practices be established to promote interoperability. This technical report describes best practices for validating KML and for using metadata and attribution. These practices further enable the goals of interoperability and increased data sharing for situation awareness across organizations and systems.

Acknowledgements

Reviewers:

- Dr. Kevin Cabana
- Curtis Brown
- Kevin Kelly
- Harry Sleeper
- Dr. Lou York

Editor:

- Katie Packard

Special thanks to:

- Tom Butts
- Thomas Fearing
- Andrew Gregorowicz
- Mark Hadley
- Todd Hay
- Dr. Mark Kramer
- Tony Lefebvre

Table of Contents

1 Introduction 1
2 Validating KML 1
2.1 Types of Validation 3
2.1.1 Well-Formed Document Validation 3
2.1.2 Schema Validation 4
2.1.3 Specification Validation 4
2.2 KML Validation Tools and Services 5
3 KML Attribution/Contribution 6
3.1 Adding Attribution as XML Comments 6
3.2 Adding Attribution to the Description 7
3.3 Adding Attribution as ExtendedData 8
4 Conclusion 9

List of Figures

Figure 1. Example of attribution as an XML comment 7
Figure 2. Example of attribution added to the description..... 7
Figure 3. Example of attribution added to the ExtendedData element 8
Figure 4. Example of attributes in ExtendedData as shown in Google Earth™..... 9

1 Introduction

Keyhole Markup Language (KML), now owned by the Open Geospatial Consortium (OGC), has evolved from a proprietary data format for Google Earth™ to an open data standard, and it has been incorporated into numerous commercial and government systems. KML can be used as a visualization and presentation data format, much like HyperText Markup Language (HTML) is used for textual data. In addition, the Extensible Markup Language (XML) underpinnings of KML and its extensible structure for user-defined metadata allow it to be used as a data-exchange and transport format for all kinds of geospatial data. Data expressed in the KML format can support a variety of needs, including emergency response, public health and safety, military command and control, fleet management, precision farming, and environmental management.

Data published in the KML format needs to be interoperable in applications and systems, including Google Earth™. The IEEE defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged."[1] In a white paper, the OGC stated that "non-interoperability impedes the sharing of data and the sharing of computing resources, causing organizations to spend much more than necessary on data, software, and hardware," adding that "organizations today are under 'economic constraints.'"[2] Therefore, limited budgets require that best practices be adopted in order to maximize cost savings.

Best practices are needed to effectively use KML as a data-exchange format. The recommendations in this report describe the need for KML validation and proper use of metadata and attribution in support of increased interoperability and data sharing. KML metadata attribution was introduced in Todd Hay's *Best Practices* white paper,[3] and this report expands on that idea with concrete examples.

2 Validating KML

The importance of validating KML is demonstrated by the rapid growth of applications other than Google Earth™ that consume and produce KML. Google has implemented extreme flexibility in its tools relative to KML in order to enable easy and widespread use of Google Earth™. However, from a validation perspective, this flexibility equates to lax validation of KML so it can accept KML loosely, adhering to the OGC KML Specification[4] and corresponding KML XML Schema.[5] This follows Postel's Law and the general principle of robustness, which is to "be conservative in what you do, and liberal in what you accept from others." Users can turn on error checking, but Google Earth™ may only prompt them if an unknown tag/namespace is encountered or the KML file is an invalid XML file (i.e., not well formed). In the early days of the World Wide Web, Web browsers such as Mosaic and Netscape allowed any HTML; they did their best to display the contents and to render any HTML tags

[1] Institute of Electrical and Electronics Engineers, 1990, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY.
[2] Reichardt, Mark, 2004, "The Havoc of Non-Interoperability: An Open GIS Consortium White Paper," Open GIS Consortium, http://portal.opengeospatial.org/files/?artifact_id=5097&version=3&format=pdf.
[3] Hay, Todd, 2006, "Best Practices - KML Reflectors/Files," The MITRE Corporation.
[4] Wilson, Tim, 2008, "KML 2.2 Specification," OGC Document: OGC 07-147r2, Open Geospatial Consortium, http://portal.opengeospatial.org/files/?artifact_id=27810.
[5] Open Geospatial Consortium, 2008, "XML Schema Document for OGC KML version 2.2," http://schemas.opengis.net/kml/2.2.0/ogckml22.xsd.

(correct context and matched begin-end tags not required). This practice encouraged content authors not to validate their content, leading to confusion when the results appeared differently in other Web browsers. Google Earth™ adopts a similar approach in how it renders KML; by default, most errors are ignored. Having data appear in Google Earth™ may give a false sense that the same data will appear like that elsewhere.

In some cases, Google Earth™ generated KML that violated the KML specification. Most of these bugs are subsequently fixed, but often not until many months after they are first discovered, resulting in a large amount of KML content being published and archived that do not conform to the standard. Generated data—bad or otherwise—tends to hang around longer than intended. Adopting a plan to reduce or eliminate bad KML early is recommended. Google has recently increased the scope of its internal beta testing program with a group of Google Earth™ "power" users to help catch bugs such as these earlier in the process.

See the following bugs for examples:	*Date reported*	*Date fixed*
• http://code.google.com/p/earth-issues/issues/detail?id=359	Jun 2009	Jul 2010
• http://code.google.com/p/earth-issues/issues/detail?id=365	Jun 2009	Aug 2011
• http://code.google.com/p/earth-issues/issues/detail?id=507	Oct 2009	--
• http://code.google.com/p/earth-issues/issues/detail?id=1182	Apr 2011	--

Although most bugs like this have been fixed, they left a legacy of invalid files on the Internet as shown in the figure below. The survey results show that most KML content found on the public Internet does not validate against the KML specification. Similar results have been seen in KML found in private and corporate networks as well.

This author conducted a survey of KML/KMZ (zipped KML files) content across the Internet in April 2011 using custom Google searches and automated tools. This resulted in a collection of 2,078 files (1,043 KML files and 1,035 KMZ files) randomly distributed among many organizations from websites around the world. The results break down as follows:

97 (5%) resulted in 404 Not Found errors or miscategorized non-KML content
1,456 (70%) failed to validate to KML Schema
525 (25%) valid with respect to KML Schema

namespace	*total*	*% total*	*# valid (% valid)*
http://earth.google.com/kml/2.0	229	11%	62 (27%)
http://earth.google.com/kml/2.1	389	19%	184 (47%)
http://earth.google.com/kml/2.2	459	22%	77 (17%)
http://www.opengis.net/kml/2.2	783	38%	198 (25%)
no default namespace or other	121	6%	4 (3%)

Of the valid 25% or 529 files, more than half had at least one violation against the KML specification, so less than 13% of the random sample was fully compliant.

Bad KML may render correctly in Google Earth™ but may fail to load or view correctly in some of the many applications that import KML and KMZ files. Developers of such applications must implement software to the strict KML standard because that is what is documented. Developers

do not have the resources or the time to test every possible case, so their environment requires more rigid interfaces, which in turn demand more adherence to specifications.

If a given KML file is used solely by its author and is not shared with others, then using Google Earth™ to view and validate will suffice. But as soon as a given KML file is accessible via a URL or is emailed elsewhere, its use is uncertain. In such cases, Google Earth™ is not the only consumer, and additional validation tools are necessary. It is likely that large systems making data available in KML will be consumed by multiple applications unknown to the data provider.

For time-critical data supporting natural disasters, national security, homeland security, airline safety, etc., using the standard correctly is essential. During natural disasters such as the 2011 Tōhoku earthquake or Hurricane Katrina is not the time to learn that KML data is not correct and that critical decisions have been made based on incorrect geo-referenced data. Consider the impact of non-interoperable data on services such as rescue operations, power, communication, and transportation, and consider the impact on safety and on repair costs.[2] Valuable time and resources spent on data conversion and translation could have been spent on validating the data conforms to the standard in the first place. A relevant discussion of the costs and issues associated with interoperability can be found in a National Institute of Science and Technology report.[6] Therefore, the benefits of data standards such as KML can be achieved by following a set of validation steps for verification.

2.1 Types of Validation

Three levels of KML validation apply to XML-based standards: well-formed document validation; schema validation; and specification validation.

2.1.1 Well-Formed Document Validation

Well-formed document validation verifies that a given XML document is well formed with respect to the XML specification, meaning that constraints such as the following are met:

- The XML document has a single root element.
- All start and end tags are matched and names are case sensitive.
- Attribute names in elements are unique (i.e., no duplicate names).
- Content contains only characters allowed by the appropriate encoding (e.g., UTF-8, etc.).

In most cases, a KML document will not load in Google Earth™ if it is not well formed.

A common error found in KML files involves labeling the file with the default UTF-8 encoding in the XML prolog while inadvertently copying/pasting an international character or graphic symbol (e.g., the degrees symbol º) from an email or document in the description field of one of the placemarks. Google Earth™ ignores such errors, but strict XML parsers fail to parse them and abort loading these documents with a fatal parse error. Replacing the *UTF-8* encoding with *ISO-8859-1* resolves this issue.

> Tip: To quickly verify whether a KML document is well formed, rename the file with an .xml extension, then drag/drop the file onto a Web browser (e.g. Internet Explorer, Firefox, etc.). Most Web browsers validate well-formed XML documents by default and will report an error, including the line number, if the XML file is not well formed.

[6] Brunnermeier, S.B., S.A. Martin, 1999, "Interoperability Cost Analysis of the U.S. Automotive Supply Chain," http://www.rti.org/publications/abstract.cfm?pub=1390.

2.1.2 Schema Validation

Schema validation occurs when an XML document conforms to declared XML Schema specifications and when its structure is valid as defined below:

- The schema defines a strict order of elements and spelling of element/attribute names, which are case sensitive.
- The KML schema defines the hierarchy of abstract and concrete elements and, as such, what elements/attributes are inherited.
- The KML schema defines where non-KML namespace elements may appear.

Valid KML documents conform to the published OGC KML 2.2 XML Schema and additional schema namespaces as defined in the given KML document instance.

Most KML documents do not conform to the strict KML XML Schema, yet they are rendered as expected in Google Earth™ clients. The majority of schema errors are out-of-order elements, which are easily identified if schema validation is being used. The problem is that other applications parsing non-compliant KML may not make the same assumptions and interpretations as Google Earth™, and they either fail to parse or, worse, misinterpret the data without displaying any error messages or prompts. Examples of the latter are presented in the next section.

2.1.3 Specification Validation

Specification validation is when KML documents conform to the strict design and business rules as defined in the specification, which includes conforming to the KML Schema. However, many of the rules are not covered in KML XML Schema validation. Additional conformance rules and best practices are described in the "OGC KML 2.2 – Abstract Test Suite."[7] In other words, a KML resource can be valid according to the KML XML Schema but still fail to conform to the specification.

For example, the KML specification defines the following rules:

- *LineString* geometries consist of two or more coordinate tuples, each containing floating point values for longitude, latitude, and altitude. The altitude component is optional. Space must be inserted between tuples. Do not include spaces within a tuple.
- *LinearRing* contains four or more tuples, each consisting of floating point values for longitude, latitude, and altitude. The altitude component is optional. Do not include spaces within a tuple. The last coordinate must be the same as the first coordinate. Coordinates are only expressed in decimal degrees.

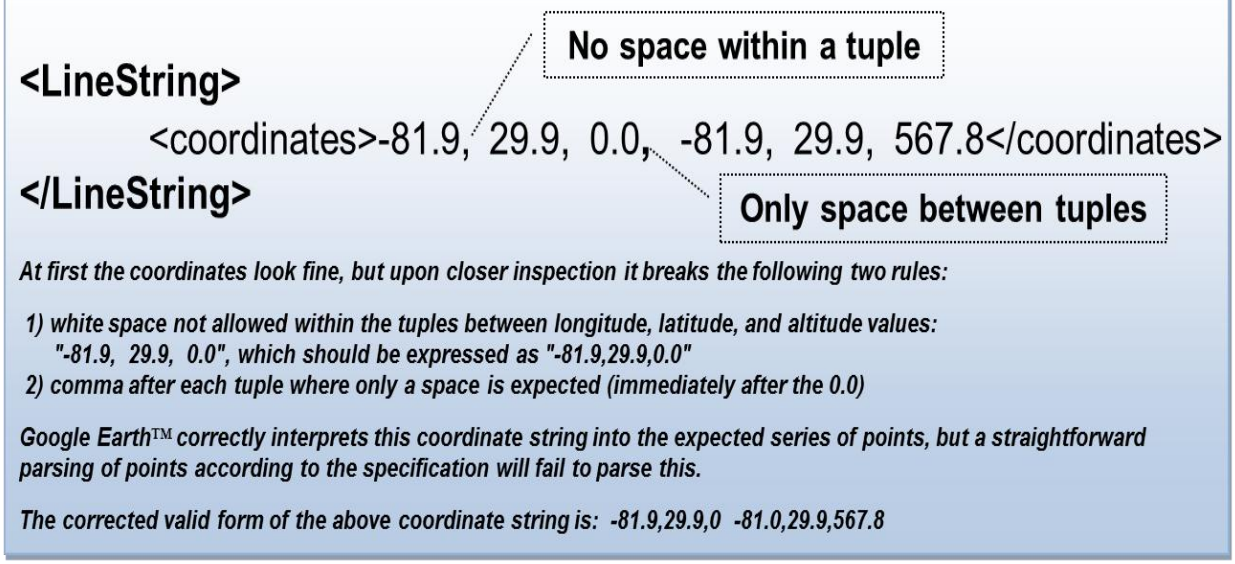
The *coordinates* component of a *LineString* and *LinearRing* from the KML XML Schema perspective must only be a list of strings separated by white space; for example, "a b c" is valid according to the KML schema definition, but not to the KML specification.

Another example is a coordinate string with the value “-81.9 29.9 0.0,” which omits the commas. At first glance, the value appears to be a single-point location at longitude=-81.9, latitude=29.9, and elevation=0. However, the specification states that tuples are separated by spaces, so in the context of a line, Google Earth™ interprets this as 3 tuples (or 3 sets of coordinates), each with the latitude and elevation missing, resulting in a coordinate list equivalent to “-81.9,0,0 29.9,0,0 0,0,0.” Google Earth™ allows the latitude component to be omitted, although it should be treated

[7] Hagemark, B., R. Martell, J. Parr-Pearson, T. Wilson, 2008, "OGC KML 2.2 – Abstract Test Suite," OGC Document: OGC 07-134r2, Open Geospatial Consortium, http://portal.opengeospatial.org/files?artifact_id=27811.

as an error since the specification requires an explicit latitude to follow each longitude value in a tuple—only the altitude component is optional.

As a real-world example, the following KML was found in a multi-agency production system that provides a live feed in KML for aircraft locations. A KML line as a point on the ground to the aircraft was in the following form:



At first the coordinates look fine, but upon closer inspection it breaks the following two rules:

1) white space not allowed within the tuples between longitude, latitude, and altitude values: "-81.9, 29.9, 0.0", which should be expressed as "-81.9,29.9,0.0"
2) comma after each tuple where only a space is expected (immediately after the 0.0)

Google Earth™ correctly interprets this coordinate string into the expected series of points, but a straightforward parsing of points according to the specification will fail to parse this.

The corrected valid form of the above coordinate string is: -81.9,29.9,0 -81.0,29.9,567.8

A KML file displaying correctly in Google Earth™ is not proof enough that the KML conforms to the KML specification. Even if a KML file validates to the KML XML Schema, the file can still not conform to the specification, as shown in the example presented above. The majority of KML content publicly available today does not strictly conform to the standard. Most of the content may render on other applications that import KML, but it could render incorrectly depending on how "bad" the KML is defined.

2.2 KML Validation Tools and Services

The following websites validate KML by URL or file upload:

- http://www.kmlvalidator.com/home.htm (allows file upload)
- http://www.feedvalidator.org/

These online tools validate KML one file at a time, which is useful for debugging. However, validation often needs to scale to large numbers of KML resources, and it should be automated in a test framework. These tools also fail to differentiate between errors and warnings, often reporting a long list of errors.

For example, The MITRE Corporation has created an internal KML library called GIScore that both parses arbitrary KML and generates valid KML. (MITRE will publicly release the GIScore software within the year.) A test suite has been created to test the parsing library against a large collection of KML samples that represent both common and uncommon cases. MITRE also developed a standalone command-line Java tool, XmlValidate. This tool validates any XML file or URL—including KML—against its target XML Schema. It uses a validating XML parser to check for well-formedness errors. The tool allows bulk validation given a target directory, and it also forces a target XML Schema other than the Schema defined in the declared namespace (e.g., validate KML 2.2 documents against KML 2.1 Schema). Additional validation tools, which test some of the KML specification "business" rules that are not caught in a Schema-only validation, have been created. Schematron, for example, is complementary technology that can validate

complex relationships and patterns in XML documents that are not possible with schema validation techniques. Therefore, a complete validation test suite is a multistep process that verifies the well formedness, schema, and specification aspects of validation.

3 KML Attribution/Contribution

In the previous section, validation was discussed to promote interoperability. This dealt with the structure of the content and whether it conformed to the specification. Next we address adding metadata and attribution to enhance the KML data, providing context with clues to what the data is, how it was collected, and links back to the data provider.

Organizations should use a disciplined approach in how they deliver their KML services. Without such an approach, information within the KML files can become outdated or used inappropriately, resulting in decisions based on misleading or obsolete information. Attribution such as the following should be included within the KML data to ensure that user expectations are clearly defined with regard to the data provided via KML:[8]

- Information Current As Of [Date]
- Frequency of KML File/Service Updates
- Originating organization or Office of Primary Responsibility
- Contact Phone Number/Website/Email Address
- Source of Information
- Method of data collection (raw-data observations, interpolation, etc.)
- Classification of Data
- Releasability of Data

Note that the exact set of metadata fields depends on the particular information domain, for example, where classification and releasability are pertinent to a military context, but not to a civilian one such as environmental management. The metadata can be added using three methods: 1) by adding the data as XML comments in the KML file, 2) by adding the text to the description of the first KML feature, and/or 3) by adding the data as ExtendedData defined in the outermost/first KML Feature. These methods can be combined; they are not mutually exclusive. The best practice is to use the second and third methods.

3.1 Adding Attribution as XML Comments

The simplest but least desirable method is to include the metadata as unformatted text nested inside an XML comment at the start of the KML file as shown in Figure 1. The attribution metadata is included with minimal change to the KML, but the data is hidden from the casual user. To view this text, an end user would need to view the source of the KML as XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
        Information Current As Of:  2010-01-25T08:00:43Z
        Frequency of KML file/service updates:  KML updated daily at 0800Z
        or something like NetworkLink auto-refreshes with default update rate at 60 second interval.
        Data updates are available every hour so clients should change the update interval to no
        less than every 60 minutes.
```

[8] Hay, Todd, 2006, "Best Practices - KML Reflectors/Files," The MITRE Corporation.

```
        Office of Primary Responsibility:  MITRE
        Contact Phone Number / Website / Email Address:  123-555-1212 /  anyuser@mitre.someorg
        Source of Information:  XYZ System
        Classification of Data:  U
        Releasability of Data:  FOUO
-->
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
...
</Document>
</kml>
```

Figure 1. Example of attribution as an XML comment

3.2 Adding Attribution to the Description

The second method is to add metadata attribution to the description element of the outermost container element in the KML. This allows producers to add attribution metadata to a KML Feature, where the description is displayed in a pop-up balloon in Google Earth™ when an end user clicks on that feature. Producers can format the description using the richness of HTML, including JavaScript and CSS. Consider adding a snippet with a short description that appears in the menu list view while the full description appears in the Feature's pop-up balloon.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
 <Snippet>
   Short description of the data/service
 </Snippet>
<description>
<![CDATA[
Description of the data/service

<B>Information Current As Of 2010-01-25T08:00:43Z</B><BR>
<B>Frequency of KML file/service updates:</B><BR>
 NetworkLink auto-refreshes with default update rate at 60 second interval.<BR>
 Data updates are available every 10 seconds so clients can change the update interval to no less than 10
seconds.<BR>
<B>Office of Primary Responsibility:</B> MITRE<BR>
<B>Contact Phone Number / Website / Email Address:</B><BR>
 123-555-1212 / anyuser@mitre.someorg<BR>
<B>Source of Information:</B> XYZ System<BR>
<B>Classification of Data:</B> U<BR>
<B>Releasability of Data:</B> FOUO
]]>
</description>
...
</Document>
</kml>
```

Figure 2. Example of attribution added to the description

3.3 Adding Attribution as ExtendedData

The third approach involves adding the metadata in an *ExtendedData* element of the KML document. This method is a machine-parsable implementation; the other methods added metadata as unstructured free-form text. Figure 3 below shows the raw KML text for this approach. As in the second approach, adding attribution to the description, in this method the end user can view the metadata formatted in a pop-up dialog box when it is viewed in Google Earth™ (see Figure 4). If the Document element has no description tag, as shown in Figure 4, then the *ExtendedData* is autoformatted in Google Earth™ as an HTML table with a border around the table cells. The *ExtendedData* element also can be inserted into a custom description or BalloonStyle using Entity Replacement for Extended Data Elements, with the names of the *ExtendedData* elements as placeholders (e.g. $[ContactPhone]).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
   <ExtendedData>
     <Data name="Information Current As Of">
       <value>2010-01-25T08:00:43Z</value>
     </Data>
     <Data name="Update Frequency">
       <value>NetworkLink auto-refreshes with default update rate at 60 second interval.
 Data updates are available every 10 seconds so clients can change the update interval to no less than 10
seconds.</value>
     </Data>
     <Data name="Office of Primary Responsibility">
       <value>MITRE</value>
     </Data>
     <Data name="ContactPhone">
       <value>123-555-1212</value>
     </Data>
     <Data name="ContactEmail">
       <value>anyuser@mitre.someorg</value>
     </Data>
     <Data name="Source">
       <value>XYZ System</value>
     </Data>
     <Data name="Classification">
       <value>U</value>
     </Data>
     <Data name="Releasability">
       <value>FOUO</value>
     </Data>
   </ExtendedData>
...
</Document>
</kml>
```

Figure 3. Example of attribution added to the ExtendedData element

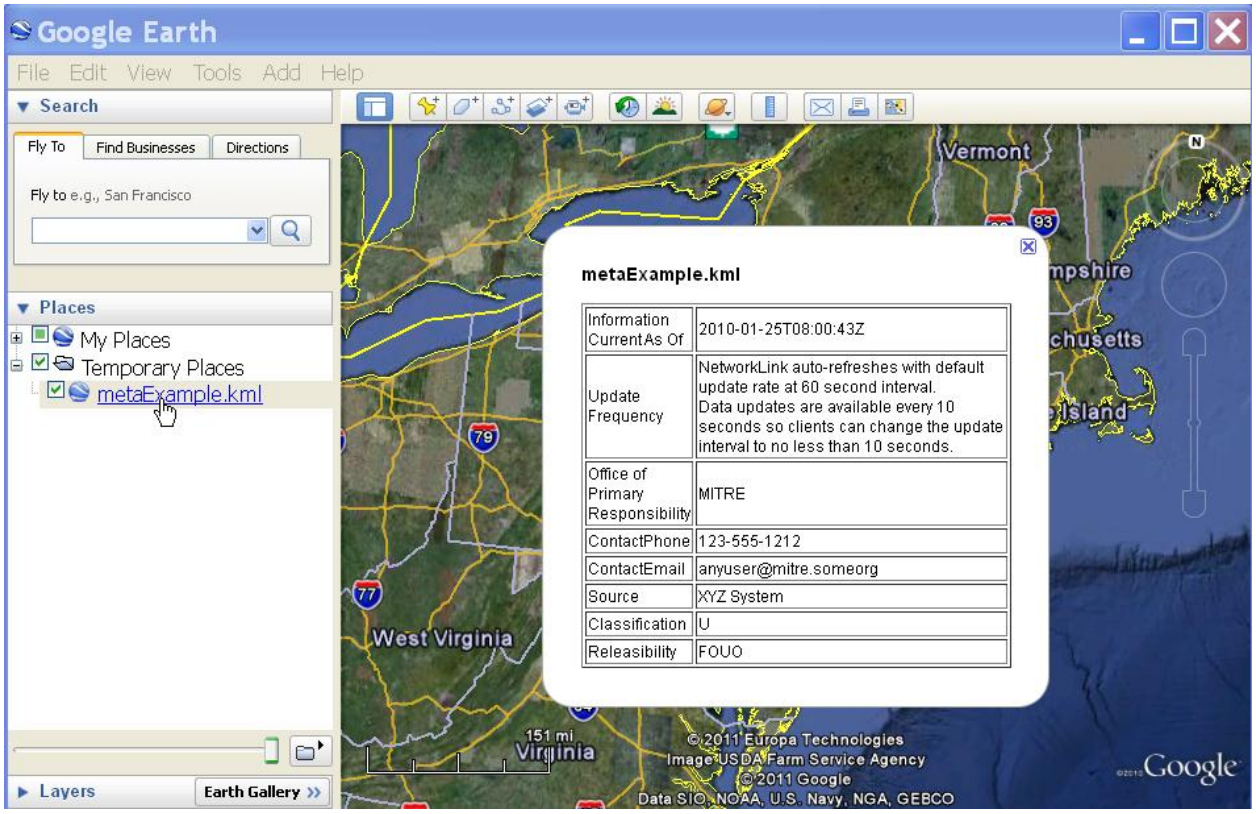


Figure 4. Example of attributes in ExtendedData as shown in Google Earth™

See Google's tutorial, "Adding Custom Data to KML,"[9] for more information on adding and formatting custom data in KML.

4 Conclusion

In order to maximize the benefits of using an open data standard, best practices such as validation and attribution are needed. Deviating from said practices reduces interoperability and increases the costs of sharing and using data, for example, by putting critical or time-sensitive missions at risk due to misleading or obsolete information. Specifically for KML, validation includes tests for well formedness, schema, and specification verification. A complete validation test suite is a multistep process that verifies these aspects. Having both human- and machine-readable metadata can better support data sharing when KML is used for data exchange and for transporting geospatial data from system to system. Metadata attribution should be included in all KML products, preferably in both the outermost description element and an extended data tag. The cost of performing validation and adding metadata attribution is small compared to the costs of not doing it and of being at risk for data exchange failures. Data not conforming to the KML standard may not show up at all or may render incorrectly. Incorrect situation awareness information could lead to wrong decisions with unintended consequences. The decision is not deciding whether to validate or not, but rather how much validation is needed.

[9] Google Earth™, "Adding Custom Data to KML," http://code.google.com/apis/kml/documentation/extendeddata.html, accessed March 2011.