Google Earth KML 2.0

Introduction

Welcome to the Google Earth KML documentation. KML, or Keyhole Markup Language, is an XML grammar and file format for modeling and storing geographic features such as points, lines, images, and polygons for display in the Google Earth client. This document describes version 2.0 for the Google Earth client version 3.0. (Earlier versions of KML can be read into the Google Earth client and saved out as KML version 2.0.)

Use KML 2.0 to:

- Specify icons and labels to identify locations on the planet surface
- Create different camera positions to define unique views for each of your features
- Use image overlays attached to the ground or screen
- Define styles to specify feature appearance
- Write HTML descriptions of features—including hyperlinks and embedded images
- Use folders for hierarchical grouping of features
- Dynamically fetch and update KML files from remote or local network locations
- Deliver current view details from the client to the server in order to fetch KML data based on changes in the 3D viewer

A KML file is processed by the Google Earth client in a similar way that HTML (and indeed, XML) files are processed by web browsers. Like HTML, KML has a tag-based structure with names and attributes used for specific display purposes. Thus, the Google Earth 3D viewer acts as a browser of KML files.

This section provides an overview of KML with the following:

What's New in KML 2.0

The following summarizes the latest features in KML version 2.0:

- A new object, <NetworkLinkControl>, controls the behavior of network files fetched through network links.
- The default format for files written by Google Earth is KMZ. This is a compressed KML file that can be opened by any software that understands common ZIP file formats. The benefit of this is that images are self-contained in the file and do not need to be hosted on a network server, as with KML 1.0.
- The <description> tag now supports a vast array of standard HTML features, such as anchors, images, rich text, and tables.
- The old <View> tag has been replaced with a new <LookAt> tag.
- The syntax of the <ScreenOverlay> feature has been updated.
- Styles are now much more powerful. They consist of styles for each geometry currently supported by the client: <IconStyle>, <LabelStyle>, <LineStyle>, and <PolyStyle>.
- Lines can now be optionally tessellated over terrain.
- The <opacity> tag has been deprecated in favor of using <color>.
- Network links are now able to return the current view to the server, allowing the server to return location-specific KML files.

KML Basics

The Google Earth client supports a number of display features for rich presentation of GIS data. All of the features supported by the Google Earth client are described in KML, which you can author a number of ways, depending upon your purpose:

- **Use the Google Earth client to author your data**

  To create KML for distribution by email or to host as non-dynamic content on a web server, use the Google Earth client. Simply structure the data the way you want it in the Google Earth client GUI and save it as a separate file (KMZ). This is also a useful way to determine the best structure for more complex KML documents you might author programmatically. The Google Earth client can create and save all but a few KML tags.

- **Use a text editor**

  You can use a text editor to create KML as a way to create simple file prototypes, or to test out the validity of a particular KML document syntax or structure.

  TIP: For quick KML syntax checking, right-click on a placemark or overlay in the Google Earth client and select Copy from the pop-up menu. You can then open a simple text document and paste the KML contents into it for a quick and easy way to view KML generated by the Google Earth client.

- **Use a development environment to generate KML programmatically**

  You can use your favorite development environment to author KML content for delivery via the web, much in the same way that other dynamic web content is created. Most content editors that display XML data can do the same for KML.

A Simple KML File

The following is an example of a simple KML file:

```
<kml xmlns="http://earth.google.com/kml/2.0">

<Placemark>
  <description><![CDATA[<a href="http://www.google.com/">Google
Search!</a>]]></description>
  <name>Google Headquarters</name>
  <LookAt>
    <longitude>-122.0839</longitude>
    <latitude>37.4219</latitude>
    <range>540.68</range>
    <tilt>0</tilt>
    <heading>3</heading>
  </LookAt>
  <Point>
    <coordinates>-122.0839,37.4219,0</coordinates>
  </Point>
</Placemark>
</kml>
```

In this example, a <kml> element contains a single <Placemark> element named *Google Headquarters*. When loaded into the Google Earth client, the <Placemark> is displayed as the default icon positioned at the longitude and latitude given by the <coordinates> element. The <LookAt> element of the <Placemark> specifies the "camera" view when displaying the placemark in the Google Earth 3D viewer.

Note how the <description> text contains HTML tags and a hyperlink. The viewer displays the description and formats it according to the HTML tags and will open clicked hyperlinks in a web browser window. (In this case, the Google homepage will open).

You can test the example KML above by copying and pasting it to a simple text editor and saving it as a file with the extension .kmz. Drag and drop the file over the 3D viewer of the Google Earth client to view the results.

Placemarks

Placemarks are the primary means of marking a location on the earth. In the Google Earth client, a placemark appears as a list item in the Places pane. It typically has an icon associated with it that marks a point on the earth; however, a placemark can also be associated with a path, a polygon, or a 3D shape. Thus, when you create a placemark, you can use the following attributes and features:

- **Geometric shapes**

  A location can be described by a number of geometric features, so a placemark can contain a single point, line, polygon, or an arbitrary collection of these features.

- **Location and altitude**

  For each of the geometric features of a placemark, you can define the location coordinates as well as the altitude of the geometry, any extrusion, and its relationship to the ground. For example, your placemark might consist of a path that you position 2km above the elevation of the terrain and have extruded so that the path appears as a fence when viewed from the side.

- **Default icon**

  You can further identify a placemark by setting a default icon for the placemark, which is used in the *Places* listing of the Google Earth client panel as well as in the 3D viewer when the placemark contains point geometry.

- **Appearance**

  Finally, you can control the appears of the placemark in the Google Earth client using a number of common KML elements as well as styles. For example, you can set the name (which appears as a label in the 3D viewer), description, default "camera" view, and default visibility for a placemark. You could then define a color and size to the label for the placemark.

Geometry

The Google Earth client supports a number of two-dimensional geometric shapes for display in the 3D viewer. These shapes are:

- **Points**

  Points can be mapped on the earth's surface as either icons or labels, or both. Placemarked points can be richly annotated within the Google Earth client, and be positioned at differing altitudes.

- **Lines**

  Line geometry includes lines and linestrings (polylines), but not road data. As with points, lines can be positioned at altitude.

- **Polygons**

  Polygons can be created in the Google Earth client. Depending upon whether you extrude the object, it can be planar or 3D. They can be completely solid, or defined with inner boundaries to create complex shapes with different styles, such as you might do for a given territory that contains lakes within. As with lines, polygons can be positioned at altitude.

You can manipulate the appearance of geometry using the following techniques:

- **Define coordinates**

  Coordinate values for geometry must be entered in degrees (longitude, latitude) and meters (elevation above sea level). KML uses the <coordinates> tag to describe this feature. See the entry for <coordinates> for more information.

  Important: The required geographic coordinates specification is Longitude, Latitude, and Altitude, in that order. If you don't follow this method, your coordinate specifications will be misinterpreted.

- **Extrude to produce 3D shapes**

  Points, lines, or polygons can be extruded, first by positioning the point at an specified altitude and then using the <extrude> element. Using a combination of points, lines, and polygons, it is easy to create rich, visually arresting displays of your data. For simple placemarks, points can be extruded to create a line that "tethers" the placemark to the ground. The extruded elements themselves can be stylized with color and line thickness, depending upon the type of geometry extruded. For lines and polygons,

- **Group into collections**

  You can group lines and polygons into collections using the <Multigeometry> tag. This feature is a useful organizational tool when you are creating three-dimensional objects consisting of numerous shapes. For example, once you have defined a geometry collection, you can easily toggle its visibility in the Google Earth client.

Image Overlays

You can use images in your KML data to create visually rich presentations, such as a GIF image of a site construction plan over the region planned for construction, or an image that serves as a legend to describe the symbols used in your KML presentation. There are two kinds of image overlays:

- **Ground Overlays**

  Ground overlays are images that are fixed to the planet surface. Use ground overlays for information-rich site display such as phased plans or weather maps. In addition to the common elements, ground overlays use the following:

  <LatLonBox>tag to define the north, south, east, and west boundary coordinates for the overlay.

  Refresh parameters that you can set to load dynamically changing imagery, such as a weather map, or imagery that loads based on the view contained in the 3D viewer.

- **Screen Overlays**

  A screen overlay is an image independent of the camera position and is fixed to the screen. Examples of screen overlays might include displaying a legend, or perhaps copyright messages. As with ground overlays, you can define multiple screen overlays and associate them with a folder. You can define the position of the screen overlay as an absolute coordinate position in the viewer, or as a percentage of the viewer size. You can also set the color of a screen overlay. See the entries for <overlayXY>, <screenXY> and <size> for more information.

  Note: Screen overlays can currently only be authored in KML using a text editor, not in the Google Earth client.

Both types of image overlays are referenced within the <Icon> tag and referred to using the <href> tag, whether the location of the file is located on a web server, a network server, or on your local machine. The following example shows a reference to a screen overlay legend on a local hard drive.

```
<Icon>
<href>C:\Documents and Settings\HP\My
Documents\Google\GoogleEarth\reLegend.gif</href>
</Icon>
```

Keep in mind that remote users do not need a local copy of an overlay image if the image is supplied remotely. In this way, you can provide real-time updates of overlay data, such as weather, or development progress of site plans. The icon tag can also contain child elements such as <refreshInterval>, which may be used to control automatic updates of remotely-served images (webcams, for example).

When you use image overlays, you can use the following features:

- **File types**

  You can use JPG, PNG, GIF, or TIFF as the file type for your overlay.

- **Draw order**

When using overlapping overlays, you can use the <drawOrder> tag to indicate the overlay stacking order. In this way, for example, overlay images with black boundary edges or masked boundaries can be assigned a lower number than adjacent overlays so as not to obscure useful information.

- **Rotation**

  You can use the <rotation> tag on the overlay image. The image can be rotated up to 180 degrees in a clockwise (positive) or counter-clockwise (negative) rotation from north.

Styles

KML styles can be employed to define the appearance of geometry, icons, balloons, and labels in the viewer. This typically includes color, size, and opacity of the feature. Similar to the behavior of styles in HTML and CSS, styles in KML can either be defined locally within the element being affected, or referred to by reference ID and reused among many elements. As with HTML, it is more efficient to design classes of styles up front and refer to them as needed, using local styles sparingly for simple, one-off effects.

Styles apply to the following elements:

- Icons
- Labels
- Lines
- Polygons—the surface of extruded lines are considered polygons
- Description Balloon

Style Effects

KML styles support the following effects:

- **Color and opacity**

  You can change the color and opacity for all elements listed above. Opacity is defined as the first hexadecimal pair of the <color> tag. For example, a 50% opaque, pink label would be defined as:

  ```
  <color>7fffaaff</color>
  ```

  When you style an icon with color, the color value is added to any existing color of the icon image itself. For this reason, color styling has the most predictable effect on white or greyscale icon imagery. Color styling applied to lines not only affects paths on the earth of lines extruded from a point, but any polygon that is drawn with outlines turned on. The color and width of the polygon's outline is derived from the line style.

- **Scale**

  Scale (<scale>) applies only to labels and icons as a floating point value with 1.0 as the original dimension of the icon image itself, or the default display size of the label. For icons, scaling is applied equally on both axes of the image.

- **Width**

  Width (<width>) applies only to lines and is defined in pixels. In addition to lines drawn as a path or extruded up from a point on the earth, width also affects outlined polygons. For example, if you are using an outlined polygon, the color and width of the polygon's outline is derived from the line style.

- **Random Color**

  You can randomize the color of all elements using the <colorMode> tag with the value set to *random*. This chooses color from the default palette at random.

- **Fill and Outline**

  These effects apply only to polygons and can be used singly or in combination. You can create a polygon structure filled with one color and outlined with a different color. Outlined polygons derive the style for the outline from the <LineStyle> feature.

- **Customized Descriptions**

  You can use the <BalloonStyle> tag and its children to create a customized title and description as well as set the color for the text itself. Typically, the description title is derived from the label text, but you can either or both, if desired.

Style Referencing

You can create styles by hand using the style features in the Google Earth client, but if you want to generate sophisticated 3D models or other data with multiple style types, you should understand how to use style referencing in order to create the most efficient KML for faster display and fewer parsing errors.

This section covers the two basic types of style referencing.

Locally Referenced Styles

You can define a style locally within the actual placemark itself. Just as with CSS styles, locally referenced KML styles need only to define their settings, because reference to the parent placemark is implicit and no selector (ID) is necessary. The following example shows a simple path with a local color style applied.

```
<kml xmlns="http://earth.google.com/kml/2.0">

<Placemark>
  <name>Google Earth - New Path</name>
  <Style>
    <LineStyle>
      <color>ff0000ff</color>
    </LineStyle>
  </Style>
  <LineString>
    <tessellate>1</tessellate>
    <coordinates>
-85.10427072849437,46.73819606924774,0 -
85.08405670114024,46.72192334662796,0
    </coordinates>
  </LineString>
</Placemark>
</kml>
```

This type of referencing works well for 3D modeling convenience when you want many elements to reference a style, but a very few to deviate from the shared style. For the most part though, local styles should be avoided where possible. When you have many local styles within placemarks, the Google Earth client must interpret and process the values defined in each style. So, for performance reasons, styles referenced by ID are strongly recommended, since the Google Earth client will only need to process a given style set once even though it might be referenced by multiple placemarks.

Styles Referenced by ID (Shared Styles)

When you use complicated styles to define multiple elements, such as icons, lines, and polygons, create a parent style at the top of the document, with all of the elements defined within it. Once you do this, any one of the placemarks can reference the style by ID, and if it contains features that map to the defined styles, the styles are applied. In this way, you do not have to have duplicate style information for the same types of geometry, and all relevant geometry can share the same style. In addition, you can contain all of your style information in an external file, and reference the entire document via the <styleUrl> tag. To create a separate style sheet, use the KML syntax as usual in that file, beginning and ending with the <Style> tag. Save the file with an extension of .xml.

The following KML example shows a document that contains two placemarks, one defining a polygon and one defining a path. Each of these placemarks use the <styleUrl> tag to refer to a collection of styles by ID defined for the parent style.

```
<styleUrl>#myDefaultStyles</styleUrl>
```

refers to

```
<Style id="myDefaultStyles">
```

which contains a number of style definitions for icons, lines, and polygons. If the entire <Style> element were contained in a separate file and hosted on a web server, the elements could reference the style by both URL and ID, such as:

```
<styleUrl>http://www.test.com/exampleStyle.xml#myDefaultStyles</styleUrl>.
```

The first placemark is a polygon with line and fill that match the polygon and line styles defined. The second placemark is a path, so its reference to the parent style matches only the line style defined. You can try the KML in your Google Earth client to see the results.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
  <!-- Begin Style Definitions -->
  <Style id="myDefaultStyles">
    <IconStyle id="khIconStyle791">
      <color>a1ff00ff</color>
      <scale>1.399999976158142</scale>
        <Icon>
          <href>root://icons/palette-4.png</href>
          <x>128</x>
          <y>64</y>
          <w>32</w>
          <h>32</h>
        </Icon>
    </IconStyle>
    <LabelStyle id="defaultLabelStyle">
      <color>7fffaaff</color>
      <scale>1.5</scale>
    </LabelStyle>
    <LineStyle id="defaultLineStyle">
```

```
      <color>ff0000ff</color>
      <width>15</width>
    </LineStyle>
      <PolyStyle id="defaultPolyStyle">
        <color>7f7faaaa</color>
        <colorMode>random</colorMode>
      </PolyStyle>
  </Style>
  <!-- End Style Definitions -->
  <!-- Placemark #1 -->
  <Placemark>
    <name>Google Earth - New Polygon</name>
    <description>Here is some descriptive text</description>
    <styleUrl>#myDefaultStyles</styleUrl>
     <Polygon id="khPolygon577">
      <extrude>1</extrude>
      <tessellate>1</tessellate>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing id="khLinearRing578">
          <coordinates>
-85.08054479600936,46.74684774665595,1840 -
85.03550980268059,46.75771393277484,1840
-85.03947773061398,46.73244936995838,1840 -
85.06072644383477,46.728009518373,1840
-85.08054479600936,46.74684774665595,1840
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark>
  <!-- Placemark #2 -->
  <Placemark>
    <name>Google Earth - New Path</name>
    <styleUrl>#myDefaultStyles</styleUrl>
    <LineString>
      <tessellate>1</tessellate>
      <coordinates>
-85.10427072849437,46.73819606924774,0 -
85.08405670114024,46.72192334662796,0
      </coordinates>
    </LineString>
  </Placemark>
</Document>
</kml>
```

Style Maps

A style map is a convenience mechanism used to define *normal* and *highlighed* states for icons in the 3D viewer of Google Earth. When an end user clicks on an icon in the 3D viewer, and that icon is correctly mapped to a state, the icon itself changes style to reflect the definitions in the style map.

The following example shows how it is possible to implement a simple "rollover" icon change. Two style IDs are defined—*normalPlacemark* and *highlightedPlacemark*. These styles are then referenced inside the <styleMap>, which defines the normal and highlighted states via a <styleUrl>. The Google Earth client uses the <key> tag to reference the appropriate <Pair> element. The placemark itself has no local style, but instead references the style map, which in turn references the appropriately defined global styles.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
  <name>Simple Icon Rollover</name>
  <open>1</open>
  <Style id="normalPlacemark">
    <IconStyle>
      <Icon>
        <href>C:/green_icon.jpg</href>
      </Icon>
    </IconStyle>
  </Style>
  <Style id="highlightPlacemark">
    <IconStyle>
      <Icon>
        <href>C:/red_icon.jpg</href>
      </Icon>
    </IconStyle>
  </Style>
  <StyleMap id="exampleStyleMap">
    <Pair>
      <key>normal</key>
      <styleUrl>#normalPlacemark</styleUrl>
    </Pair>
    <Pair>
      <key>highlight</key>
      <styleUrl>#highlightPlacemark</styleUrl>
    </Pair>
  </StyleMap>
  <Placemark>
    <name>Example Placemark</name>
    <styleUrl>#exampleStyleMap</styleUrl>
    <Point>
      <coordinates>-
```

```
122.0856545755255,37.42243077405461,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

Grouping Mechanisms

A KML file can contain more than one placemark, as illustrated in some of the KML examples in Style Referencing. In addition, a single placemark can contain more than one type of geometry, such as both a point and a polygon. As your data set grows in complexity, you can also use folders to organize placemarks, overlays, and even other folders. In order to achieve this, the following grouping mechanisms are available in KML:

Documents

A document is a root-level container to hold all KML hierarchies, including style and schema elements and their children in addition to all other features, such as folders, placemarks, and overlays. When you open a document with styles and multiple placemarks in the Google Earth client, it can be expanded to view the contents just as you would a folder. A single KML file will have only one <Document> tag.

Note: When you use global styles or schemas for your data, you must also use the <Document> element to contain these elements along with all the features. Even if your KML file contains only one feature, but also has a schema or a global style definition, you must define those within a document structure.

Folders

Most uses of KML employ folders to organize placemarks, screen and ground overlays, and other folders. Use folders to structure collections of features and overlay groups, and to provide a unified view for a group of placemarks and/or overlays. You can define multiple folders within folders much in the same way you nest tables in HTML.

In addition to the common elements, folders can have the <open> property, which is a switch (0 or 1) that controls whether or not the folder will be in an expanded or collapsed state when it is loaded.

Geometry Collections

You can use the <MultiGeometry> element to group a number of geometry collections within a single placemark, such as when creating a 3D model for display in a specific region. You can define multiple geometry collections and toggled them on or off in the *Places* listing in the Google Earth client. This is convenient when you create particularly complex models, and you want to visualize parts of the model on screen or troubleshoot your design.

Network Links

Use network links to quickly and easily share KML files among multiple users by sharing a *link* to the data rather than distributing the data itself. You can think of a network link as a folder that has attributes that you can define and then place on a server (either a local area network or a web server). This enables users to easily obtain dynamic data contained in the network link. For example, you can collect a number of weather satellite imagery overlays, along with other data that is polled from other servers, and bundle all the information together in a network link that people can open to view weather forecast models superimposed on the earth.

Network links can use all the common elements as well as most the features available to folders. The rest of this section describes the features specific to the network link.

Location

Use the <href> element to indicate the location of the KML or script that you are linking to as the network link.

Note: When using to a local area network path, the required KML syntax for path separation is a forward-slash (//) rather than the Microsoft double back-slash (\).

Refresh Features

The network link provides a number of refresh features that lets you define whether the network link should refresh the data, how often, and if the data should refresh when the view changes. The following features can be set:

- **Refresh interval**

  Use the <refreshMode> to set the network link to update either on document load or on periodic intervals. Use the <refreshInterval> to indicate the time in seconds to refresh the network link when you have set the mode to *interval*.

- **View-based refresh**

  A view-based query can be returned to the server either periodically, according to the view refresh interval, or after a set time has elapsed since the "camera" has stopped moving. When active, the Google Earth client sends an HTTP GET to the source server with the coordinates of the view window appended as a bounding box (BBOX) value. In this way, the source server may then return information in the context of the active view.

  You can also set the data boundary relative to the view using the <viewBoundScale> element. By default, there is a 1:1 relationship between the bounding box of the view and data that Google Earth requests from the server, so the value is set to 1.0, but you can set the value to .75, for example, to scale the amount of displayed data to 75% of the view.

- **Fly to View on Refresh**

  The <flyToView> element is a switch control that automatically updates the view to the new coordinates in the <LookAt> element when those are updated by the server. You can use this feature to control the viewer behavior, such as when you want to provider a "server-side tour" of a particular set of features, or when you want to define the ability to pull the viewer to the new coordinates in the event of an emergency.

Network Control Features

There are a number of features in the <NetworkLinkControl> element that allow link creators to control the behavior of network links from the server side. See <NetworkLinkControl> for details.

Common Elements

KML features share a number of common elements, described below.

- **Name**

  The KML syntax uses the <name></name> tag to provide the name of the folder, placemark, overlay, or other features for easy identification in the Google Earth client. The following example shows the <name> element applied to a <Folder> element.

```
<Folder>
  <name>My House</name>
  .
  .
  .
</Folder>
```

- **Description**

  Use the <description> element to enter additional information about the feature. The Google Earth client intelligently recognizes valid web URLs entered in the description and automatically provides a link in the description without the need for HTML markup. However, if you desire, you can format the appearance of the descriptive text using HTML to provide font styling and even images.

  In order to use any HTML markup in the description of an element, you will need to use the CDATA XML element to avoid XML parsing of special characters, or you will need to provide HTML entity references for those individual characters. The following example illustrates the CDATA element.

```
<description><![CDATA[This is a description that contains
<em>HTML formatting</em> for line breaks<br /> and simple
font styling. ]]></description>
```

- **Visibility**

  Use the <visibility> tag to set the default visibility of the feature when the KML file is first opened in the Google Earth client. (After opening the KML file, the user always has the option to toggle the display of the feature on or off.) Visibility applies to all features. However, visibility has special conditions when used on features and overlays contained by folders. In that case, visibility of a feature is always determined by the feature itself, not by the container of the feature.

- **View coordinates**

Use the <LookAt> element to specify the view around a given point on the planet surface. The Google Earth client can set the "camera" view for placemarks, folders, overlays, and other features. When a feature has a specified <LookAt> tag, double-clicking the item in the Places window causes the Google Earth client to pan or tilt to achieve the specified view.

Important: Remember that the view for a feature bears no relationship to the coordinates or extent (longitude/latitude) of the feature. For example, a placemark might be positioned in the center of a large plot of land, and the observation coordinates for that placemark might look off toward the left corner of the land rather than directly down over the placemark itself.

In fact, the view for a placemark can be set so that the geometry itself is not even in view. Therefore, when you set a view for a placemark, do not assume that it automatically keeps the placemark in the Google Earth client viewer. You can use this behavior you want to mark a "camera view" of a particular position that is not associated with any geometry; for example, a view of the north rim of the Grand Canyon. You could even set a placemark in California and set the view for that placemark in New York. Because of this, use this feature carefully to avoid user confusion or undesired behavior. When a feature has no <LookAt> tags, double-clicking the feature causes the Google Earth client to zoom to a view directly over the feature.

- **Snippet**

  Use the <snippet> tag to provide a brief description that appears beneath the feature when that feature is displayed in the *Places* pane in the Google Earth client. Use this tag to create your own brief description when you want to override the appearance of the <description> element's content in the *Places* pane.

Schemas

For elements in your KML that are not native to the default KML schema, you can define your own schema as a set of named and typed XML elements. For example, you can specify how KML should interpret imported files such as from a GPS tracker. A schema requires the following:

- **Parent element**

  The KML entity that the new schema will inherit from. For example, if you want the data in your KML document to inherit the elements of a placemark, you would set Placemark as the parent. See the entry for <parent> for more information.

- **Schema name**

  As with all other features, the name element identifies the schema. See the entry for <name> for more information.

- **Field declaration(s)**

  The field declaration specifies the individual fields in your KML data and defines their name and type.

The following example illustrates:

```
<Schema>
  <name>States</name>
  <parent>Placemark</parent>
  <SimpleField>
    <name>FIPS</name>
    <type>wstring</type>
  </SimpleField>
  <SimpleField>
    <name>STATE</name>
    <type>wstring</type>
  </SimpleField>
</Schema>
```

Here, the schema defines two simple fields --FIPS, and STATE -- both of which are wide string fields (unicode UCS2 encoding). The schema uses the Placemark element as its parent and so inherits all the attributes of the Placemark element. Subsequent use of Placemark fields in the KML will be correctly interpreted by the Google Earth client as elements similar in structure to Placemark elements.

KML Syntax Rules

As with any programming or scripting language, KML has rules of grammar and syntax that must be followed.

Document Structure

The structure of a KML document obeys all of the syntax rules of XML. This document assumes familiarity with basic XML syntax rules, but the following list provides the high-level guidelines.

- XML tags must always have an opening and closing tag
- XML tags are case sensitive
- For KML, tags with initial upper case characters are complex tags, while tags using initial lower case characters are simple tags. See the entry for Tag Types for details.
- XML elements must be properly nested and well formed (see http://www.w3.org/TR/xhtml1/#h-4.1)
- An XML document must always have a single root element

  For KML, this means you can use the <kml></kml> tag, the <Document></Document> tag, the <Folder></Folder> tag, or even the <Placemark></Placemark> tag as the root
- Attribute values must always be quoted
- CR/LF is converted to a new line (in HTML description, this translates to
)
- Comments in XML are similar to comments in HTML

For more detail, consult XML reference manuals or search online for XML syntax guidelines. If the structure of your KML document does not follow the basic XML formatting rules, an error dialog in the Google Earth client indicates that a formatting error occurred. (See http://www.w3.org/XML/ for more information on XML.)

Note: XML Schema validation is not yet enabled in the Google Earth client.

Tag Types

There are two basic types of KML tags: simple and complex. Complex tags are easily identified by an initial upper case letter, while simple tags use all lower case. Complex tags can function as parent tags to both complex and simple tags, while simple tags are children only and can contain no other tags.

String Values

Any string value in KML, such as strings used in the name and description tags, can be entered in UTF-8 encoding of Unicode, and this will be properly interpreted by the Google Earth client.

KML Tag Dictionary

This section contains a reference for all of the usable KML tags in alphabetical order. Each section lists the tag name with its correct case, its type, values if any, and the containment hierarchy of the tag. The closing tag is not listed in the header, but its use is assumed.

<address>

This tag can contain an address written as a standard Street, City, State address, or as a postal code. You can use the <address> tag to specify the location of a point instead of using latitude and longitude coordinates. For example, the Google Earth client can compute the position of

```
<address>1600 Amphitheater Pkwy, Mountain View, CA</address>
```

Note: This feature currently works only for U.S., Canada, and United Kingdom addresses

Values

A string value representing the street address or postal code of the desired placemark. For example:

```
<address>1600 Amphitherter Pkwy, Mountain View, CA</address>
```

Parents

Contained by:

<Placemark>

Children

None.

<altitudeMode>

Modifies the altitude for a placemark or other type of geometry. When using shared or referenced styles, you can use this tag for folders as well. When altitude for geometry is set as a positive value above zero, the altitude mode determines the geometry's relationship to the ground based on one of three values.

Values

Can use one of the following values:

- *clampedToGround*

  This is the default altitude mode for elements when not otherwise specified and when no altitude is provided in the <coordinate> tag.

- *relativeToGround*

  Sets the altitude of the element relative to the actual ground elevation of particular location. If the ground elevation of a location is exactly at sea level, and the altitude for a point is set to 9 meters, then the placemark elevation is 9 meters with this mode. However, if the same placemark is set over a location where the ground elevation is 10 meters above sea level, the elevation of the placemark is then 19 meters.

- *absolute*

  Sets the altitude of the element exactly above sea level, regardless of the actual elevation of the terrain beneath the element. For example, if you set the altitude of a placemark to 10 meters with an absolute altitude mode, the placemark will appear to be at ground level if the terrain beneath is also 10 meters above sea level. If the terrain is 5 meters above sea level, the placemark will appear elevated above the terrain by 5 meters.

Parents

Contained by:

<Point>

<LineString>
<Polygon>

Children

None.

<color>

A tag representing color that can be applied to any geometry. Color values are expressed in hexadecimal notation, including opacity (alpha) values. The order of expression is alpha, blue, green, red (ABGR). The range of values for any one color is `0` to 255 (`00` to `ff`). For opacity, `00` is fully transparent and `FF` is fully opaque.

Note: When color is applied to an icon, the texture color of the icon is multiplied by the <geomColor> value. For example, if your icon's image is a bluish green color, and you set a greenish yellow color for the placemark, the result will be green. Keep in mind that with multiply blend mode, RGB values are multiplied component wise (R*R, G*G, B*B). Consequently, pure red (1, 0, 0) times pure green (0, 1, 0) yields black, because (1*0, 0*1, 0*0) is (0, 0, 0). For this reason, best effects for adding color to icons are achieved with greyscale icons.

Values

The standard range from `00000000` to `ffffffff`. For example, if you want to apply a blue color with 50% opacity to an icon, your code would look similar to the following:

```
<IconStyle>

   <color>7fff0000</color>
   <Icon>
     <href>root://icons/palette-3.png</href>
     <w>32</w>
     <h>32</h>
   </Icon>
 </IconStyle>
```

Parents

Contained by:

<PolyStyle>

<LineStyle>

<IconStyle>

Children

None.

<colorMode>

Used to set a color mode of randomized or standard. You can use this tag in conjunction with lines, icons, polygons, or labels. When you enable referencing of a single style effect a folder (see Style Referencing), then the color mode chose applies to all elements in the folder that reference that style. In the Google Earth client, this is referred to as *style sharing*.

Values

There are only two values for this element: *normal*, and *random*. By default, if <colorMode> is not specified, the a normal color mode is applied. When you apply a random color mode, a randomized color is chosen and applied to the base color and opacity. This means that if you desire a fully randomized color effect, you should set your base color to white (`<color>ccffffff</color>`). If your base color is greyscale, the randomized color is added to that value, so you can achieve a more muted tone to the randomized color by setting the base color to a greyscale value. The darker the greyscale, the darker the randomized color.

Finally, if you choose any other color besides white or greyscale as your base color value, then the randomization applies a gradient to the base value, resulting in various shades of the base color in the resulting geometry. For example, if you choose red as the base color for the polygon style, and then set the color mode to random, the polygons that reference that style will appear in various shades of red. The following example shows a polygon style with those settings.

```
<PolyStyle>
  <color>ff0000ff</color>
  <colorMode>random</colorMode>
</PolyStyle>
```

In this example, the opacity is set at 100%. For best effect, an opacity of 100% should be used when you are using a random color mode applied to a single base color; otherwise, the shade variations between elements is not as evident.

Parents

Contained by:

<PolyStyle>

<LineStyle>

<IconStyle>

Children

None.

<cookie>

Use the <cookie> element to append the text to the URL query on the next refresh of the network link. You can use this in your script to provide more intelligent handling on the server side, including version querying and conditional file delivery.

```
<cookie>someCookieText</cookie>
```

Values

A user-specified string.

Parents

<NetworkLinkControl>

Children

None.

<coordinates>

Defines the exact coordinates of the point location in longitude, latitude, and altitude—in that precise order. Values are separated by commas. Multiple coordinates are separated by a space. Longitude and latitude measurements are standard lat-lon projection with WGS84 datum. Google Earth uses simple cylindrical projection (or Plate Carée), which is a simple map projection where the meridians and parallels are equidistant, straight parallel lines, with the two sets crossing at right angles. The following snippet shows a point with longitude and latitude with no elevation specified.

```
<Point>
  <coordinates>-111.661,33.2212,0<coordinates>
<Point>
```

Values

Determined by the position of the point coordinates. The value is expressed in decimal degrees and meters above sea level.

Important: The required geographic coordinates specification is Longitude, Latitude, and Altitude, in that order. If you don't follow this method, your coordinate reference will be inaccurate.

Parents

Contained by:

<LinearRing>

<LineString>

<Point>

Children

None.

<description>

Supplies descriptive information that appears in the information balloon when the user clicks on either the placemark name in the Places pane, or the placemark icon. This text also appears beneath the placemark in the Places pane if no <snippet> tag is specified for the feature.

The description element supports plain text as well as HTML formatting. A valid URL string for the World Wide Web is automatically converted to a hyperlink to that URL (e.g. http://www.google.com). Consequently, you do not need to surround a URL with the `<a href="http://.."></a>` tags in order to achieve a simple link. However, if you enter a description in plain text, extra white space and carriage returns are ignored. Use the HTML
 element for line breaks, or fully format your description in HTML.

When using HTML to create a hyperlink around a specific word, or when including images in the HTML, you must use HTML entity references or the CDATA element to escape angle brackets, apostrophes, and other special characters. The CDATA element tells the XML parser to ignore special characters used within the brackets. This element takes the form of:

```
<![CDATA[ special characters here ]]>
```

If you prefer not to use the CDATA element, you can use entity references to replace all the special characters. However, it is most often simplest to use the CDATA element to enclose the entire HTML string rather then to substitute entity references for each special character.

The following example illustrates the description tag used to generate the pop-up illustrated above:

```
<description><![CDATA[This is an image <img src="C:\Documents
and Settings\My Documents\Icons\icon.gif"> and we have a link
http://www.google.com.]]></description>
```

Values

User-defined.

Important: The description element supports HTML formatting only. It does not support other web-based technology, such as dynamic page markup (PHP, JSP, ASP), scripting languages (VBScript, Javascript), nor application languages (Java, Python).

Parents

Can be contained by:

<Document>

<Folder>

<NetworkLink>

<GroundOverlay>

<ScreenOverlay>

<Placemark>

Children

None.

<Document>

The root element for KML documents. It also acts as a folder for the elements it contains. This tag is required if your KML file uses schemas, shared styles, or contains more than one feature, such as multiple placemarks. For a single feature using a local style, it is not required.

Values

None.

Parents

None.

Children

Can contain the following tags:

<description>

<Document>

<Folder>

<GroundOverlay>

<name>

<LookAt>

<NetworkLink>

<Placemark>

<ScreenOverlay>

<visibility>

<drawOrder>

Use this tag when defining overlapping overlays to determine the stacking order for the images. The default value is 0, so if no <drawOrder> tag is specified for the overlay, it's draw order is set to that. Overlays with a higher draw order values are drawn on top of overlays with lower draw order values.

```
<GroundOverlay>
  <name>Overlay2</name>
  <drawOrder>1</drawOrder>
  <Icon>
    <href>http://www.test.com/images/projectStart.jpg</href>
    <viewBoundScale>0.75</viewBoundScale>
  </Icon>
  <LatLonBox>
    <north>37.06467534059571</north>
    <south>37.02453087540336</south>
    <east>-121.7764114763516</east>
    <west>-121.9034130727271</west>
  </LatLonBox>
</GroundOverlay>
<GroundOverlay>
```

Values

Range from 0 to 99.

Parents

Contained by:

<GroundOverlay>

<ScreenOverlay>

Children

None.

<east>

This tag defines the longitude of east edge of the overlay image.

```
<LatLonBox>
  <north>37.06467534059571</north>
  <south>37.02453087540336</south>
  <east>-121.7764114763516</east>
  <west>-121.9034130727271</west>
</LatLonBox>
```

Values

Dependent on the actual position required for the overlay image. You can specify longitude values as decimal degrees (WGS84).

Parents

Contained by:

<LatLonBox>

Children

None.

<extrude>

Allows vertical extrusion of two-dimensional features. Placemarks, paths, and polygons can be extruded to form three-dimensional objects. See Geometry for more information.

```
<Polygon>

   <extrude>1</extrude>
   <tessellate>1</tessellate>
   <altitudeMode>relativeToGround</altitudeMode>
   <outerBoundaryIs>
       . . .
   </outerBoundaryIs>
 </Polygon>
```

Values

A Boolean value of 0 for false and 1 for true.

Parents

Contained by:

<LineString>

<Polygon>

<Point>

Children

None.

<fill>

A tag representing whether or not Google Earth renders a polygon with a fill (a solid color of varying opacity). If set to true, the fill color and opacity are derived from the <color> tag for the polygon. If neither <fill> nor <outline> are specified for the polygon, the polygon is drawn with *both* fill and outline. To set only fill for a polygon, you can simply set <outline> to `0`.

Values

A Boolean value of 0 for false and 1 for true. The default is true.

Parents

Contained by:

<PolyStyle>

Children

None.

<flyToView>

A boolean tag that is a child of <NetworkLink>. When set, updated KML from the server causes the viewer to update to the current view. This should be used sparingly for best user experience, but is effective for such situations as alerts or notifications where the current 3D view in the client needs to be overridden.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
</NetworkLink>
</kml>
```

Values

Boolean, `0` for off, 1 for on. The default value is off if the tag is not specified.

Parents

<NetworkLink>

Children

None.

<Folder>

A top-level, optional tag used to structure hierarchical arrangement of other folders, placemarks, ground overlays, and screen overlays. Use this tag to structure and organize your information in the Google Earth client. See the introductory entry for Grouping Mechanisms for more details.

```
<Folder>

   <name>Name of Folder</name>
   <description>Descriptive text</description>
      <Folder>
        <name>SubFolder #1 Name</name>
        <description>Descriptive text</description>
        <Placemark>
          [placemark data here ...]
        </Placemark>
      </Folder>
      <Folder>
         <name>SubFolder #2 Name<name>
         <description>Descriptive text<description>
         <Placemark>
            [placemark data here ...]
         <Placemark>
      <Folder>
<Folder>
```

Values

None.

Parents

Can be contained by the following:

<Document>

<Folder>

<NetworkLink>

Children

Can contain the following tags:

<description>

<Document>

<Folder>

<geomColor>

This tag is deprecated. Use <color> tag instead.

<GeometryCollection>

This tag is deprecated. Use <MultiGeometry> instead.

<geomScale>

This tag is deprecated. Use <scale> instead.

<GroundOverlay>

This element contains tags for defining and placing an overlay image on the globe. See Image Overlays for more information. The following example shows an overlay image for a weather map that is rotated in the 3D viewer and set with a slight transparency. By default, the visibility is set to on for ground overlays unless specified otherwise.

```
<GroundOverlay>
  <name>Weather Map</name>
  <LookAt>
    <longitude>-90.86879847669974</longitude>
    <latitude>48.25330383601299</latitude>
    <range>440.8490922646644</range>
    <tilt>8.39474026454335</tilt>
    <heading>2.701112047774894</heading>
  </LookAt>
  <color>9effffff</color>
  <drawOrder>1</drawOrder>
  <Icon>
    <href>http://www.example.com/weatherMap.png</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>3600</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewBoundScale>0.75</viewBoundScale>
  </Icon>
  <LatLonBox>
    <north>48.25475939255556</north>
    <south>48.25207367852141</south>
    <east>-90.86591508839973</east>
    <west>-90.8714285289695</west>
    <rotation>39.37878630116985</rotation>
  </LatLonBox>
</GroundOverlay>
```

Values

None.

Parents

Can be contained by:

<Document>

<Folder>

Children

Can contain the following:

<color>

<drawOrder>

<Icon> (required)

<LatLonBox> (required)

<LookAt>

<name>

<visibility>

<h>

When extracting portions of an image palette for use as an icon, <h> is the height of the extracted portion of the image in pixels. The following code snippet shows how the Google Earth client uses the <h> value with an icon palette to define a region that serves as a single icon for a placemark.

```
<Icon>
  <href>root://icons/palette-3.png</href>
  <x>192</x>
  <y>96</y>
  <w>32</w>
  <h>32</h>
</Icon>
```

Use this tag in conjunction with all the tags in the example above to properly define an icon region.

Values

An integer representing the height of the extracted portion in pixels.

Parents

Contained by:

<Icon>

Children

None.

<heading>

When used as a child of <LookAt>, <heading> describes the angular distance along the horizon to the viewpoint. This is measured from north. The following example shows a heading due west.

```
<LookAt>
  <longitude>-90.86879895668633</longitude>
  <latitude>48.25329704862119</latitude>
  <range>436.2888293403289</range>
  <tilt>8.394733469490348</tilt>
  <heading>-90</heading>
</LookAt>
```

When <heading> is used as a child of <IconStyle>, it will fix the orientation of the icon so that it always points north, regardless of the heading of the view. Ordinarily, the icon has a fixed orientation relative to the camera position, so as the viewer is rotated, the icon always points to the top of the screen. Using the <heading> tag, you can fix the position of the icon uses a value in degrees from north. To give an example, this is how you would rotate the "down arrow" to point to the right (note that the orientation of the icon is now absolute, and will rotate with the camera view):

```
 <IconStyle>
 <heading>270</heading>
 <Icon>
   <href>root://icons/palette-4.png</href>
   <x>128</x>
   <y>128</y>
   <w>32</w>
   <h>32</h>
 </Icon>
 </IconStyle>
```

Values

User defined, set it degrees of rotation (0 -360).

Parents

Contained by:

<IconStyle>

<LookAt>

Children

None.

<href>

Defines the location of the image to be used as the overlay or as the icon for the placemark. This location can either be on a local file system or a remote web server.

```
<Icon>
  <href>http://www.example.com/weatherMap.png</href>
  <refreshMode>onInterval</refreshMode>
  <refreshInterval>3600</refreshInterval>
  <viewRefreshMode>onStop</viewRefreshMode>
  <viewBoundScale>0.75</viewBoundScale>
</Icon>
```

Values

Defined by the location of the image file. For example, an image on a local C drive might be defined as:

```
C:/GoogleEarth/example.jpg
```

If you specify an image on a web server, specify the entire URL for that image:

```
http://www.example.com/images/maps/weatherFrance.png
```

You can also use an image palette and specify a portion of the image for your icon. See <Icon> for more information.

Parents

Is contained by:

<Icon>

Children

None.

<Icon>

Defines an image associated with a style or overlay. The required <href> child element defines the location of the image to be used as the overlay or as the icon for the placemark. This location can either be on a local file system or a remote web server.

```
<Icon>
  <href>root://icons/palette-3.png</href>
  <x>128</x>
  <y>128</y>
  <w>32</w>
  <h>32</h>
</Icon>
```

In addition to referring to a single image as an icon, you can also use a palette syntax for a larger image to "extract" a portion of the image to serve as the icon. A palette image typically consists of a single image that is a composite of smaller images intended to represent single icons. Palette images use texture memory more efficiently, so consider using palette images for improved performance on the Google Earth client.

Using the palette syntax, you can refer to a specific portion of that image for individual icons. To use this syntax, you need four coordinates for the portion of the master image you want to use:

- x coordinate, or position from the left most edge, in pixels
- y coordinate, or position from the bottommost edge, in pixels
- w coordinate, or width of the extracted portion, in pixels
- h coordinate, or height of the extracted portion, in pixels

In the example above, the palette is a 4x4 image grid consisting of 32-pixel icons in each cell. The cell that is chosen for the icon in this example is the one in the upper right corner.

When using your own imagery for icons or palettes, keep in mind the following suggestions:

- Length and width in pixels of an icon or image palette needs to be a power of 2
- Icon palettes larger than 256x256 pixels might not work on all video cards (those that ship with the Google Earth client measure 256 x 256)
- Individual icon images also need to have a length/width in pixels that is a power of 2
- There must be at least one row of 100% transparent pixels along each boundary of an icon, which means that adjacent icons in a palette will have 2 transparent pixels between them
- Those pixels that have 100% opacity should also have their RGB values set to black. Because of the way some graphics chips filter the icons as they are drawn, this will avoid having the color from the transparent pixels bleed into the visible ones.
- Icons generally look best if they have a distinctly colored border around them, such as black
- Icons with straight horizontal or vertical lines tend to flicker when the user zooms in or out

In addition to defining regions of icons, the <Icon> element also controls the refresh parameters for dynamically-driven images.

Values

None.

Parents

Is contained by:

<GroundOverlay>

<ScreenOverlay>

<Style>

Children

Contains:

<h>

<href>

<refreshMode>

<viewRefreshMode>

<w>

<x>

<y>

<IconStyle>

<IconStyle> specifies the following style properties for icons when drawing them in the 3D viewer:

- Color

  White and 100% opaque is default value for color (`<color>ffffffff</color>`).

- Color mode

  Normal is the assumed default mode.

- Scale

  The default scale is 1.0.

The following code snippet illustrates an icon style with a color of purple, a scale of 1.4, and a slight opacity.

```
<IconStyle>
  <color>ed7f0055</color>
  <colorMode>normal</colorMode>
  <scale>1.4</scale>
  <Icon>
    <href>root://icons/palette-3.png</href>
    <w>32</w>
    <h>32</h>
  </Icon>
</IconStyle>
```

Values

None.

Parents

<Style>

Children

Contains:

<color>

<colorMode>

<heading>

<Icon>

<scale>

<innerBoundaryIs>

Defines the inner boundary of a polygon. This can be used to define a lake region within a polygon describing a parcel area, or it can be used when modeling 3D geometry.

```
<Polygon>
   <innerBoundaryIs>
      <LinearRing>
        <coordinates>-88.306534, 30.227852, 0.000000
...... -88.306534, 30.227852, 0.000000
        </coordinates>
      </LinearRing>
   </innerBoundaryIs>
</Polygon>
```

Values

None.

Parents

Is contained by:

<Polygon>

<key>

Indicates either a normal or highlighted state for a style. See Style Maps for more details.

```
<StyleMap id="example_style">
   <Pair>
      <key>normal</key>
      <styleUrl>#example_style_off</styleUrl>
   </Pair>
   <Pair>
      <key>highlight</key>
      <styleUrl>#example_style_on</styleUrl>
   </Pair>
</StyleMap>
```

Values

Use either *normal* to indicate the style URL to be used as the default style for the feature, or *highlight* to indicate the style URL to be used when the feature is selected.

Parent Tags

Is contained by:

<Pair>

Children

None.

<labelColor>

Deprecated. Use <labelStyle> instead.

<labelStyle>

<LabelStyle> specifies the following style properties for icons when drawing them in the 3D viewer:

- Color

  White and 100% opaque is default value for color (<color>ffffffff</color>).

- Color mode

  Normal is the assumed default mode.

- Scale

  The default scale is 1.0.

The following code snippet illustrates an icon style with a color of purple, a scale of 1.4, and an opacity of 93%.

```
<LabelStyle>
  <color>ed7f0055</color>
  <colorMode>normal</colorMode>
  <scale>1.4</scale>
</LabelStyle>
```

Values

None.

Parents

<Style>

Children

Contains:

<color>

<colorMode>

<scale>

<latitude>

This element defines the distance on the Earth (measured in degrees) north or south of the equator.

```
<LookAt>
  <longitude>-90.86879847669974</longitude>
  <latitude>48.25330383601299</latitude>
  <range>440.8490922646644</range>
  <tilt>8.39474026454335</tilt>
  <heading>2.701112047774894</heading>
</LookAt>
```

Values

Determined by the observation coordinates of the particular view. Its units can be expressed in decimal degrees.

Parents

Contained by:

<LookAt>

Children

None.

<LatLonBox>

This tag is used to specify the coordinates for the overlay itself.

```
<LatLonBox>
  <north>48.25475939255556</north>
  <south>48.25207367852141</south>
  <east>-90.86591508839973</east>
  <west>-90.8714285289695</west>
  <rotation>39.37878630116985</rotation>
</LatLonBox>
```

Values

None.

Parents

Contained by:

<GroundOverlay>

Children

Contains the following tags, most required:

<east>

<west>

<north>

<south>

<rotation> (not required)

<LinearRing>

Defines the structure of a polygon, which must have an outer boundary, and can optionally have one or more inner boundaries to define a hole in the polygon. Uses the <coordinates> tag to express the values of the coordinate line strings. The final coordinate in the line marks the last point before reconnecting with the first coordinate.

```
<Polygon>
   <outerBoundaryIs>
      <LinearRing>
        <coordinates>
-88.306534, 30.227852, 0.000000
...
88.306534, 30.227852, 0.000000
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
</Polygon>
```

Values

None.

Parents

Contained by:

<innerBoundaryIs>

<outerBoundaryIs>

Children

Contains only one child:

<coordinates>

<LineString>

Defines a line. Defines the location of a string of coordinates on the map. Uses the <coordinates> tag to express the values of the points.

```
<LineString>
  <tessellate>1</tessellate>
  <coordinates>
-88.306534, 30.227852, 0.000000
...
88.306534, 30.227852, 0.000000
  <coordinates>
</LineString>
```

Values

None.

Parents

Can be contained by:

<Placemark>

<MultiGeometry>

Children

Contains only one child:

<coordinates>

<LineStyle>

Specifies the following style properties for lines when drawing them in the 3D viewer:

- Color
  White and 100% opaque is default value for color (<color>ffffffff</color>).
- Color mode
  Normal is the assumed default mode.
- Width
  The default width is 1.0 pixel.

The following code snippet illustrates a 50% opaque red line with a width of 4 pixels.

```
<LineStyle id="khLineStyle989">
  <color>7f0000ff</color>
  <width>4</width>
</LineStyle>
```

Values

None

Parents

Is contained by:

<Style>

Children

May contain:

<color>

<colorMode>

<width>

<linkDescription>

This tag allows the server to control the appearance of the Network Link's description and will override the string that appears in the Network Link's <description> field. You can use this to override description edits made by the end user.

```
<linkDescription><![CDATA[KML now has new features
available!]]></linkDescription>
```

Values

A user-specified string. See the information for <description> for details on text formatting.

Parents

<NetworkLinkControl>

Children

None.

<linkName>

Allows the server to control the appearance of the Network Link's name in the *Places* view. It will override the string that appears in the Network Link's <name> field. You can use this to override edits made by the end user.

```
<linkName>New KML features</linkName>
```

Values

A user-specified string. See the information for <name> for details on text formatting.

Parents

<NetworkLinkControl>

Children

None.

<longitude>

This element defines the distance on the Earth (measured in degrees) east (positive values above 0 to 180 degrees) or west (negative values below 0 to 180 degrees) of the Greenwich Meridian.

```
<LookAt>
  <longitude>-90.86879895745898</longitude>
  <latitude>48.25329705085941</latitude>
  <range>738.6083569292965</range>
  <tilt>8.394733471860304</tilt>
  <heading>2.701081133141738</heading>
</LookAt>
```

Values

Determined by the observation coordinates of the particular view. Its units can be expressed in decimal degrees.

Parents

Contained by:

<LookAt>

Children

None.

<LookAt>

Defines the observation coordinates or eye point of the parent placemark, folder, or ground overlay.

```
<LookAt>
  <longitude>-90.86879895745898</longitude>
  <latitude>48.25329705085941</latitude>
  <range>738.6083569292965</range>
  <tilt>8.394733471860304</tilt>
  <heading>2.701081133141738</heading>
</LookAt>
```

Values

None.

Parents

Can be contained by:

<Folder>

<Document>

<Placemark>

<GroundOverlay>

Children

Contains the following tags:

<heading>

<latitude>

<longitude>

<range>

<tilt>

<message>

When specified, the string within this tag will be displayed in a pop-up message box in the Google Earth client when the file is first fetched. It will be displayed again only if the message text has changed, or if the file is completely reloaded into the client.

```
<message>This is a pop-up message. You will only see this
once</message>
```

Values

A user-specified string.

Parents

<NetworkLinkControl>

Children

None.

<minRefreshPeriod>

Specified in seconds, this is the minimum allowed time between fetches of the file. If, for example, you have a file that is changing only once every hour, you can throttle fetches of that file to a minimum of once every hour.

```
<minRefreshPeriod>3600</minRefreshPeriod>
```

Values

Number specified in seconds.

Parents

<NetworkLinkControl>

Children

None.

<MultiGeometry>

A tag used to group more than one geometry element, such as multiple polygons used to define a single feature (such as a 3D building) for display in the Google Earth 3D viewer.

```
<MultiGeometry>
   <Polygon>
      <outerBoundaryIs>
         <LinearRing>
            <coordinates>.....
            </coordinates>
         </LinearRing>
      </outerBoundaryIs>
    </Polygon>
    <Polygon>
      <outerBoundaryIs>
         <LinearRing>
            <coordinates>.....
            </coordinates>
         </LinearRing>
      </outerBoundaryIs>
    </Polygon>
</MultiGeometry>
```

Values

None.

Parents

Can be contained by the following:

<Placemark>

Children

Can contain

<LineString>

<MultiGeometry>

<Point>

<Polygon>

<name>

This tag is used in the Google Earth client as the label for a placemark, folder, or network link, for example. It can also be used by the <Schema> element as an identifier.

```
<Folder>
  <name>Favorite Places</name>
  . . .
</Folder>
```

Values

User-defined text. HTML markup is not supported. All entered characters are displayed in the 3D viewer as the label.

Parents

Can be contained by:

<Folder>

<Document>

<GroundOverlay>

<Placemark>

<ScreenOverlay>

<Schema>

Children

None.

<NetworkLink>

Defines a referenced KML file on a local or remote network. You can set the location of the link to the KML file using the <Url> tag. Within that tag, you can define the refresh options in order to update information based on time and camera change. You can also use the <viewFormat> tag to return a variable to the server that contains a string containing client viewing information, such as what the current bounding box of the client is, or the current heading.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
<visiblity>1</visbility>
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
<refreshVisibility>1</refreshVisibility>
</NetworkLink>
</Document>
</kml>
```

Values

None.

Parents

Can be contained by:

<Folder>

<Document>

Children

Can contain the following:

<flyToView>

<name>

<refreshVisibility>

<Url>

<visibility>

<NetworkLinkControl>

Controls the behavior of files fetched via NetworkLinks. Using this feature, you have access a number of features via its children. Specifically, these include:

- **Minimum refresh period**

  You can use the <minRefreshPeriod> element as a server throttle to limit the number of fetches to your server to a specified minimum period. For example, if a user sets a link refresh to 5 seconds, you can set your minimum refresh period to 3600 to limit refresh updates to every hour.

- **Pop-up message**

  You can deliver a pop-up message, such as usages guidelines for your network link. The message appears only once each time the network link is loaded into the client, or if the message text is updated on the server.

- **Cookie**

  Use the <cookie> element to append the text to the URL query on the next refresh of the network link. You can use this in your script to provide more intelligent handling on the server side, including version querying and conditional file delivery.

- **Name and description**

You can control the name and the description of the network link from the server, so that changes made to the name and description on the server side are over-ridden by the server.

```
<NetworkLinkControl>
  <message>This is a pop-up message. You will only see this
once</message>
  <cookie>someCookieText</cookie>
  <linkName>New KML features</linkName>
  <linkDescription><![CDATA[KML now has new features
available!]]></linkDescription>
</NetworkLinkControl>
```

Values

None.

Parents

<Document>

Children

<cookie>

<linkDescription>

<linkName>

<message>

<minRefreshPeriod>

<north>

Defines the latitude of the north edge of the overlay image.

```
<LatLonBox>
  <north>48.25475939255556</north>
  <south>48.25207367852141</south>
  <east>-90.86591508839973</east>
  <west>-90.8714285289695</west>
  <rotation>39.37878630116985</rotation>
</LatLonBox>
```

Values

Dependent on the actual position required for the overlay image. Latitude value can be specified as decimal degrees.

Parents

Contained by:

<LatLonBox>

Children

None.

<ObjArrayField>

Defines a schema field for an array of (pointers to) schema objects. Uses the <type> element for the type name of the objects in the array.

Values

None.

Parents

Is contained by:

<Schema>

Children

Contains the following:

<name>

<type>

<ObjField>

Used to define a type of field (pointer to) for a schema object. Uses the <type> element for the type name of the object.

Values

None.

Parents

Is contained by:

<Schema>

Children

Contains the following:

<name>

<type>

<open>

This element defines whether the folder appears open or closed when Google Earth first loads the folder.

```
<Folder>

   <name>Name of Folder</name>
   <open>0</open>
   <description>Descriptive text</description>
      <Folder>
        <name>SubFolder #1 Name</name>
        <description>Descriptive text</description>
        <Placemark>
          [placemark data here ...]
        </Placemark>
      </Folder>
      <Folder>
         <name>SubFolder #2 Name<name>
         <description>Descriptive text<description>
         <Placemark>
            [placemark data here ...]
         <Placemark>
      <Folder>
<Folder>
```

Values

On load, a value of 1 will cause the folder to appear in its expanded state. A value of `0` will load the folder in its collapsed state. Set the folder to load in the collapsed state for folders containing a large number of features. This will ensure faster load time in the Google Earth client.

Parents

Contained by:

<Folder>

Children

None.

<outerBoundaryIs>

Defines the outer boundary of a polygon. Required.

```
<Polygon>
   <outerBoundaryIs>
      <LinearRing>
        <coordinates>-88.306534, 30.227852, 0.000000
...... -88.306534, 30.227852, 0.000000
        </coordinates>
      </LinearRing>
   </outerBoundaryIs>
</Polygon>
```

Values

None.

Parents

Is contained by:

<Polygon>

Children

Contains:

<LinearRing>

<outline>

A simple tag representing whether or not a polygon is to be rendered with an outline. If set to true, the outline color and opacity are derived from the <color> tag for the <LineStyle>. If neither <fill> nor <outline> are specified for the polygon, the polygon is drawn with *both* fill and outline. To set only outline for a polygon, you can simply set <fill> to `0`.

Values

A Boolean value of 0 for false and 1 for true. The default is true.

Parents

Contained by:

<PolyStyle>

Children

None.

<Pair>

Defines a key/value pair to provide multiple style references for such things as modes. See Style Maps for more information.

```
<StyleMap id="example_style">
   <Pair>
      <key>normal</key>
      <styleUrl>#example_style_off</styleUrl>
   </Pair>
   <Pair>
      <key>highlight</key>
      <styleUrl>#example_style_on</styleUrl>
   </Pair>
</StyleMap>
```

Values

None.

Parents

Is contained by:

<StyleMap>

Children

Must contain the following:

<key>

<styleUrl>

<parent>

Defines the base type to which further schema fields are added. See Schemas for more details.

```
<Schema>
   <name>High School</name>
   <parent>Placemark</parent>
   <SimpleField>
      <name>Address</name>
      <type>wstring</type>
   </SimpleField>
   <SimpleField>
      <name>Average SAT score</name>
      <type>int</type>
   </SimpleField>
</Schema>
```

Values

Defined as one of the existing KML base types.

Parents

Is contained by:

<Schema>

Children

None.

<Placemark>

Use a placemark to describe a point (Icon), line, path or polygon on the planet surface. You can define a number of other elements for placemark entries, including <LookAt> coordinates, name, and description. See Placemarks for more details.

```
<Placemark>
   <name>Google Earth - New Placemark</name>
   <description>Some Descriptive text.</description>
   <LookAt>
     <longitude>-90.86879847669974</longitude>
     <latitude>48.25330383601299</latitude>
     <range>440.8490922646644</range>
     <tilt>8.39474026454335</tilt>
     <heading>2.701112047774894</heading>
   </LookAt>
   <styleUrl>#myPointStyle</styleUrl>
   <Point>
     <coordinates>-
90.86948943473118,48.25450093195546,0</coordinates>
   </Point>
 </Placemark>
```

Values

None.

Parents

The <Placemark> tag can be contained by:

<Folder>

<Document>

Children

Can contain the following tags and their children, where applicable:

<description>

<LookAt>

<name>

<Point>

<styleUrl>

<Style>

<visibility>

<overlayXY>

Defines the coordinate point on the overlay image itself that will be used to map to the screen coordinate. It requires X and Y values, and the units for those values (either pixels or fraction). For example, `<overlayXY x="1" y="1" xunits="fraction" yunits="fraction"/>` affects the upper right corner of the image. Used with `<screenXY of x="-50" y="0.9" xunits="pixels" yunits="fraction"/>`, this measurement places the upper right corner of the image 50 pixels inset from the right edge of the screen and 10% below the top edge of the screen.

Values

The x and y components can be specified in one of the following ways:

Center the image

```
<ScreenOverlay>
  <overlayXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
  <screenXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Place the image on the top left

```
<ScreenOverlay>
  <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Placing the image at the right of the screen

```
<ScreenOverlay>
  <overlayXY x="1" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="1" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Parents

Is contained by:

<ScreenOverlay>

Children

None.

<Point>

Defines the location of a point coordinate on the map. Uses the <coordinates> tag to express the values of the point location.

```
<Placemark>
   . . .
   <Point>
     <coordinates>-
90.86948943473118,48.25450093195546,0</coordinates>
   </Point>
 </Placemark>
```

Values

None.

Parents

Can be contained by:

<MultiGeometry>

<Placemark>

Children

Contains only one child:

<coordinates>

<Polygon>

Defines a polygon. Uses <outerBoundaryIs> by default to define an outer boundary.

```
<Polygon>
   <outerBoundaryIs>
      <LinearRing>
        <coordinates>-88.306534, 30.227852, 0.000000
...... -88.306534, 30.227852, 0.000000
        </coordinates>
      </LinearRing>
   </outerBoundaryIs>
</Polygon>
```

You can also use <innerBoundaryIs> to define holes within the polygon.

Values

None.

Parents

Can be contained by:

<MultiGeometry>

<Placemark>

Children

Can contain:

<outerBoundaryIs>

<innerBoundaryIs>

<PolyStyle>

The <PolyStyle> element indicates the following style properties for polygons when drawing them in the 3D viewer:

- Color
- Color mode
- Fill
- Outline

By default, both fill and outline are assumed to be true if neither are specified. The following code snippet illustrates a polygon style with color, color mode, and outline but no fill.

```
<PolyStyle>
  <color>ff0000ff</color>
  <colorMode>random</colorMode>
  <fill>0</fill>
  <outline>1</outline>
</PolyStyle>
```

Values

None

Parents

Is contained by:

<Style>

Children

May contain:

<color>

<colorMode>

<fill>

<outline>

<range>

The <range> tag determines the altitude of the eye point.

```
<LookAt>
  <longitude>-90.86879895745898</longitude>
  <latitude>48.25329705085941</latitude>
  <range>738.6083569292965</range>
  <tilt>8.394733471860304</tilt>
  <heading>2.701081133141738</heading>
</LookAt>
```

Values

Determined by the observation coordinates of the particular view. Its units can be expressed in decimal degrees.

Parents

This tag is contained only by the <LookAt> tag.

Children

None.

<refreshInterval>

Indictes the time to refresh a network link url or overlay image in seconds.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<GroundOverlay>
  <name>MergedReflectivityQComposite_20051006-012008.tif</name>
  <Icon>
    <href>http://www.example.com/weather/NW/weatherMap.jpg</href>
    <viewRefreshMode>onRequest</viewRefreshMode>
  </Icon>
  <LatLonBox>
    <north>0</north>
    <south>0</south>
    <east>0</east>
    <west>0</west>
  </LatLonBox>
  <refreshInterval>60</refreshInterval>
</GroundOverlay>
</kml>
```

Values

Indicated in seconds. If the value is greater than 0, the image or URL is refreshed every n seconds. If the value is less then 0, the image will never be loaded, and if the value is equal to 0, the image is loaded once.

Parents

Can be contained by:

<GroundOverlay>

<ScreenOverlay>

<Url>

Children

None.

<refreshMode>

Sets the type of refresh that is done to a network link or ground overlay, either refreshing upon a designated interval or only once upon loading in the Google Earth client (default).

```
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
    . . .
  <Icon>
   <href>http://wdssii.nssl.noaa.gov/geotiff/NW/MergedReflectivityQComposite_20051005-
231649.tif?</href>
   <refreshMode>onInterval</refreshMode>
   <viewRefreshMode>onRequest</viewRefreshMode>
  </Icon>
```

Values

Use *onInterval* to indicate a time-based refresh of the KML or the ground overlay. Use *once* or leave the tag out to indicate refresh only upon loading of the network link or overlay into the Google Earth client.

Parents

<Icon>

<Url>

Children

None.

<refreshVisibility>

Maintains the default visibility of features within the KML document to which the network link points. When enabled, overrides visibility set by the Google Earth user when the link is refreshed.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
<visiblity>1</visbility>
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
   . . .
  </Url>
  <refreshVisibility>1</refreshVisibility>
</NetworkLink>
</Document>
</kml>
```

Values

The default value is `0`, leaving the visibility within the control of the Google Earth client. Set the value to 1 to reset the default visibility of features each time the network link is refreshed. For example, suppose a placemark within the linked KML file has <visibility> set to 1 and <refreshVisibility> set to 1. When the file is first loaded into the Google Earth client, the user can clear the check box next to the item to turn off display in the 3D viewer. However, when the network link is refreshed, the item will be made visible again, since its default visibility state was true.

Parents

Is contained by:

<NetworkLink>

Children

None.

<rotation>

Indicates the rotational axis of the image from its center point.

```
<LatLonBox>
  <north>48.25475939255556</north>
  <south>48.25207367852141</south>
  <east>-90.86591508839973</east>
  <west>-90.8714285289695</west>
  <rotation>39.37878630116985</rotation>
</LatLonBox>
```

Values

Use + or - 180 to indicate the rotation of the image from 0, which is the default orientation of the image.

Parents

Can be contained by:

<LatLonBox>

<ScreenOverlay>

Children

None.

<Schema>

Defines a custom schema to enable KML to properly interpret elements in your KML not native to the default KML schema. Using this tag and its children, you can define your own schema as a set of named and typed XML elements. See Schemas

Values

None.

Parents

None.

Children

Is parent to the following elements:

<name>

<parent>

<ObjField>

<ObjArrayField>

<SimpleField>

<SimpleArrayField>

<scale>

Scales the dimensions of an element along both x and y axes. You can use this attribute to scale icons and labels to alter their size in the Google Earth 3D viewer.

```
<IconStyle id="khIconStyle1027">
  <color>7fff0000</color>
  <scale>1.799999952316284</scale>
  <Icon>
    <href>root://icons/palette-3.png</href>
    <w>32</w>
    <h>32</h>
  </Icon>
</IconStyle>
```

Values

Values are indicated as a floating point unit. For example, since the default size of an icon is `.01`, a <scale> value of `0.5` would decrease the overall dimensions of the icon or label by half. Similarly, a <scale> value of `2.0` would double the overall dimension of the icon or label.

Parents

Contained by:

<IconStyle>

<LabelStyle>

Children

None.

<ScreenOverlay>

This element contains tags for defining and placing an image on the screen. See Image Overlays for details. The KML code for placing an image (with original width, height and aspect ratio) at the exact center of the screen looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
  <ScreenOverlay id="khScreenOverlay756">
  <description>This screen overlay uses fractional positioning to
put the image in the exact center of the screen</description>
  <name>Simple crosshairs</name>
  <visibility>0</visibility>
  <Icon>
     <href>http://myserver/myimage.jpg</href>
  </Icon>
  <overlayXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
  <screenXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
  <rotation>39.37878630116985</rotation>
  <size x="0" y="0" xunits="pixels" yunits="pixels"/>
  </ScreenOverlay>
</kml>
```

Values

None.

Parents

Can be contained by:

<Folder>

<Document>

Children

Can contain the following:

<drawOrder>

<Icon> (required)

<overlayXY>

<rotation>

<screenXY>

<size>

<visibility>

<screenXY>

Defines the coordinate points on the screen itself that the overlay image will be mapped to. For example, a screenXY of (-50, 0.9) with an overlayXY of (1,1) places the upper right corner of the image 50 pixels inset from the right edge of the screen and 10% below the top edge of the screen.

The x and y components can be specified in one of the following ways:

Center the image

```
<ScreenOverlay>
  <overlayXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
  <screenXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Place the image on the top left

```
<ScreenOverlay>
  <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Placing the image at the right of the screen

```
<ScreenOverlay>
  <overlayXY x="1" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="1" y="1" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

Parents

Is contained by:

<ScreenOverlay>

Children

None.

<SimpleArrayField>

Defines an array of simple field types like ints, floats, or strings. See Schemas for more details.

Values

None.

Parents

Is contained by:

<Schema>

Children

Can contain the following:

<name>

<type>

<SimpleField>

Defines simple field types like ints, floats, or strings. See Schemas for more details.

```
<SimpleField>
  <name>height</name>
  <type>int</type>
</SimpleField>
```

Values

None.

Parents

Is contained by:

<Schema>

Children

Can contain the following:

<name>

<type>

<size>

Specifies the size of the image of the screen overlay.

Values

Use of the size tag is best illustrated by example.

To force the image to maintain its native height, width, and aspect ratio, set the values to zero:

```
<size x="0" y="0" xunits="fraction" yunits="fraction"/>
```

To force the image to retain its horizontal dimension, but to take up 20% of the vertical screen space:

```
<size x="0" y="0.2" xunits="fraction" yunits="fraction"/>
```

To force the image to resize to 100px by 500px:

```
<size x="100" y="500" xunits="pixels" yunits="pixels"/>
```

Parents

Is contained by:

<ScreenOverlay>

Children

None.

<south>

This tag defines the latitude of the south edge of the overlay image.

```
<LatLonBox>
  <north>48.25475939255556</north>
  <south>48.25207367852141</south>
  <east>-90.86591508839973</east>
  <west>-90.8714285289695</west>
  <rotation>39.37878630116985</rotation>
</LatLonBox>
```

Values

Dependent on the actual position required for the overlay image. Latitude value can be specified as decimal degrees.

Parents

Contained by:

<LatLonBox>

Children

None.

<snippet>

A simple tag that causes a short description to be displayed in the *Places* pane beneath the placemark. By default, the <description> tag is used beneath the placemark, but you can use this tag to customize a short description instead. If a placemark contains both a description and a snippet, the snippet appears beneath the placemark in the *Places* pane, and a description appears in the description balloon.

```
<Placemark>
   <name>Google Earth - New Placemark</name>
   <description>Some Descriptive text.</description>
   <snippet>Here is some snippet in bold</snippet>
   <LookAt id="khLookAt780">
       <longitude>-90.86879847669974</longitude>
       <latitude>48.25330383601299</latitude>
       <range>440.8490922646644</range>
       <tilt>8.39474026454335</tilt>
       <heading>2.701112047774894</heading>
    </LookAt>
    <styleUrl>#khStyle721</styleUrl>
    <Point id="khPoint781">
       <coordinates>-
90.86948943473118,48.25450093195546,0</coordinates>
    </Point>
</Placemark>
```

Values

User defined text. HTML markup is not supported.

Parents

Contained by:

<Placemark>

Children

None.

<Style>

Indicates drawing style such as for custom icons, geometry scale and color, and label scale. Styles are referenced by ID placemarks or geometry, so you can share a style among many geometry or placemark elements. See Styles for more information.

Note: A style must have an ID in order to be referenced.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
  <!-- Begin Style Definitions -->
  <Style id="myDefaultStyles">
    <IconStyle id="khIconStyle791">
      <color>a1ff00ff</color>
      <scale>1.399999976158142</scale>
        <Icon>
          <href>root://icons/palette-4.png</href>
          <x>128</x>
          <y>64</y>
          <w>32</w>
          <h>32</h>
        </Icon>
    </IconStyle>
    <LabelStyle id="defaultLabelStyle">
      <color>7fffaaff</color>
      <scale>1.5</scale>
    </LabelStyle>
    <LineStyle id="defaultLineStyle">
      <color>ff0000ff</color>
      <width>15</width>
    </LineStyle>
      <PolyStyle id="defaultPolyStyle">
        <color>7f7faaaa</color>
        <colorMode>random</colorMode>
      </PolyStyle>
  </Style>
  <!-- End Style Definitions -->
  <!-- Placemark #1 -->
  <Placemark>
    <name>Google Earth - New Polygon</name>
    <description>Here is some descriptive text</description>
    <styleUrl>#myDefaultStyles</styleUrl>
    . . .
  </Placemark>
  <!-- Placemark #2 -->
  <Placemark>
```

```
      <name>Google Earth - New Path</name>
      <styleUrl>#myDefaultStyles</styleUrl>
      . . . . .
    </Placemark>
  </Document>
  </kml>
```

You can specify a <Style> within a <Placemark>. This style is known as the Placemark's local style and any fields that are specified in the local style will override the corresponding fields in the referenced style (i.e., any style referenced through <styleUrl>). Minimal use of local styles is strongly recommended for best efficiency.

Values

Style has one attribute, ID, which can have any arbitrary value. The ID is used by elements that reference the style.

Parents

Can be contained by:

<Document>

<Folder>

Children

Can own the following:

<IconStyle>

<LabelStyle>

<LineStyle>

<PolyStyle>

<StyleMap>

Use a <StyleMap> element to provide a normal and highlighted icon for a placemark, so that the highlight version appears when the user mouses over the icon in Google Earth. See Style Maps for more information and examples.

Values

The ID of the style is the attribute for this element.

Parents

Can be contained by:

<Document>

Children

Contains the following:

<Pair>

<styleUrl>

References a <Style> or <StyleMap> by a URL. For referenced style elements that are local to the KML document, a simple # referencing is used; otherwise use a full URL along with # referencing when styles are contained in external files. Examples are:

```
<styleUrl>#myIconStyleID<styleUrl>
<styleUrl>http://someserver.com/somestylefile.xml#restaurant<styleUrl>
```

Values

The ID or URL of the style to be referenced.

Parents

Can be contained by:

<Pair>

<Placemark>

Children

None.

<tessellate>

Use tessellation to allow lines and paths to follow the terrain.

```
<tessellate>1</tessellate>
```

Values

If undefined, tessellation is off. Use 1 to turn on tessellation.

Parents

Contained by:

<LineString>

Children

None.

<tilt>

This tag indicates the angle of the eyepoint to the designated point.

```
<LookAt>
  <longitude>-90.86879895745898</longitude>
  <latitude>48.25329705085941</latitude>
  <range>738.6083569292965</range>
  <tilt>8.394733471860304</tilt>
  <heading>2.701081133141738</heading>
</LookAt>
```

Values

A value of 0 indicates no tilt and the perspective in the Google earth client viewer of looking straight down on the object. A value of 90 indicates full tilt and the perspective of a horizon view in the Google Earth client. The range of values is from 0 to 90 (representing degrees of tilt).

Parents

Contained by:

<LookAt>

Children

None.

<type>

Defines the type of field declared in the schema.

```
<SimpleField>
  <name>Population<name>
  <type>int<type>
<SimpleField>
```

Values

The possible range of values for fields are:

- uint
- short
- ushort
- float
- double
- bool
- string
- wstring (This is wide string, which is UCS2 unicode)
- sharedstring
- sharedwstring
- Vec2f (A two-dimensional vector float)
- Vec3d (A three-dimensional vector double)
- icon

Parents

Can be contained by the following:

<ObjArrayField>

<ObjField>

<SimpleArrayField>

<SimpleField>

Children

None.

<Url>

You can set the location of the link to the KML file, define the refresh options for the server and viewer changes, and populate a variable to return useful client information to the server.

```
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
```

Values

None.

Parents

Is contained by:

<NetworkLink>

Children

Contains:

<href> (required)

<refreshInterval>

<refreshMode>

<viewFormat>

<viewRefreshMode>

<viewRefreshTime>

<viewBoundScale>

Used for overlays—in either network links or placemark overlays—to indicate the percentage of screen space to fill with data. If you are creating a ground overlay to deliver dynamic data via a network link, you can set the view-bound scale to 1.0 so that the dynamic date entirely fills the screen. This would be a typical setting to use when the user will not need to adjust the size of the overlay.

Values

View bound scale is set to 1.0 as default for network links and 0.75 for overlays if not specified. Otherwise, you can set the value to a fractional portion of the screen size, including values greater than 1.0.

Keep in mind that when using this tag for ground overlays where the user might want to modify the position, you should keep the value to a smaller fraction of the screen size to provide for easier editing. Otherwise, set the bound to reflect the area of data relative to the viewing boundaries of the client screen that you want to display.

Parents

<Icon>

Children

None.

<viewRefreshMode>

A child of the <Url> element for network links and the <Icon> element when used for ground overlays. Indicates the type of view-based refresh to use, it specifies how the client should return the view coordinates to the server specified in the <href> tag.

```
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
</NetworkLink>
```

Values

- *never*

  Default mode. If <viewRefreshMode> is undefined, data is not refreshed when the camera view changes in the Google Earth client.

- *onStop*

  Returns the coordinates *n* seconds after movement in the viewing window has stopped, where n is defined in <viewRefreshTime>

- onRequest

  Returns the coordinates only when the user chooses to refresh the overlay or network link

Parents

Contained by:

<Icon>

<Url>

Children

None.

<viewRefreshTime>

A simple tag and a child of the <Url> element for network links and the <Icon> element when used for ground overlays. Specifies the frequency with which to return the view coordinates to the server.

```
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
</NetworkLink>
```

Values

An integer representing seconds.

Parents

Contained by:

<Icon>

<Url>

Children

None.

<viewFormat>

A child of the Network Link's <Url> element that allows more complete control over the view information returned to the server. Without specifying this tag, the information returned is the WMS-style east, south, west, north bounding-box coordinates, but you can now return any of these parameters in whichever order you like, and also a number of parameters in the <LookAt> element. The following example returns all possible information in a comma-delimited string:

```
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Url>
    <href>http://www.example.com/geotiff/NE/MergedReflectivityQComposite.kml</href>
    <refreshMode>onInterval</refreshMode>
    <refreshInterval>30</refreshInterval>
    <viewRefreshMode>onStop</viewRefreshMode>
    <viewRefreshTime>7</viewRefreshTime>
    <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth],
[lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading]</viewFormat>
  </Url>
</NetworkLink>
```

Values

A user defined variable. Available parameters are as follows:

- [bboxWest]
- [bboxSouth]
- [bboxEast]
- [bboxNorth]
- [lookatLon]
- [lookatRange]
- [lookatTilt]
- [lookatHeading]

Parents

<Url>

Children

None.

<visibility>

Defines the default visibility of a folder, a placemark, or an overlay. If visibility is set to off, the element is loaded into the Google Earth client but does not appear in the 3D viewer until the user turns it on. The default (undefined) visibility state is on.

```
<Placemark>

   <name>My House</name>
   <visibility>0</visibility>
    . . . .
 </Placemark>
```

Values

0 sets default visibility off

1 sets default visibility on

Parents

Can be contained in:

<Folder>

<Document>

<GroundOverlay>

<Placemark>

<ScreenOverlay>

Children

None.

<w>

When extracting potions of an image for use as an icon, <w> is the width in pixels of the extracted portion of the image. See <Icon> for more information.

```
<w>32</w>
```

Values

An integer representing the width of the extracted potion of an image in pixels.

Parents

Contained by:

<Icon>

Children

None.

<west>

This tag defines the longitude of the west edge of the overlay image.

```
<LatLonBox>
  <north>48.25475939255556</north>
  <south>48.25207367852141</south>
  <east>-90.86591508839973</east>
  <west>-90.8714285289695</west>
  <rotation>39.37878630116985</rotation>
</LatLonBox>
```

Values

Dependent on the actual position required for the overlay image. Longitude value can be specified as decimal degrees.

Parents

Contained by:

<LatLonBox>

Children

None.

<width>

Indicates the width of lines in paths or polygons in pixels. The following example shows a <LineStyle> with a width of 4 pixels.

```
<LineStyle id="khLineStyle989">
  <color>7f0000ff</color>
  <width>4</width>
</LineStyle>
```

Values

Single integers indicating the width of the line in pixels. Fractional values are supported as well as values greater than the recommended range of 0 - 4 pixels. Bear in mind, however, that not all graphics cards support line width values other than the recommended range. Values larger than 4 pixels can appear to be "split" in the 3D viewer where lines bend in new directions.

Parents

Is contained by:

<LineStyle>

Children

None.

<x>

When extracting potions of an image for use as an icon, <x> is the offset from the left edge in pixels. See <Icon> for more information.

```
<x>128</x>
```

Values

An integer representing the left offset of the extracted potion of an image in pixels.

Parents

Contained by:

<Icon>

Children

None.

<y>

When extracting potions of an image for use as an icon, <y> is the offset from the bottom edge, in pixels. See <Icon> for more information.

```
<y>64</y>
```

Values

An integer representing the bottom offset of the extracted potion of an image in pixels.

Parents

Contained by:

<Icon>

Children

None.