Open Geospatial Consortium

Submission Date: 2014-11-10

Approval Date: 2015-05-18

Publication Date: 2015-08-04

External identifier of this OGC® document: http://www.opengis.net/doc/kml/2.3

Internal reference number of this OGC® document: OGC 12-007r2

Version: 1.0

Category: OGC® Implementation

Editor: David Burggraf

OGC KML 2.3

Copyright notice

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Standard
Document subtype:	Encoding Standard
Document stage:	Approved for Public Release
Document language:	English

Contents

i. Abstract

KML is an XML grammar used to encode and transport representations of geographic data for display in an earth browser. Put simply: KML encodes what to show in an earth browser, and how to show it. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard.

The KML community is wide and varied. Casual users create KML Placemarks to identify their homes, describe journeys, and plan cross-country hikes and cycling ventures. Scientists use KML to provide detailed mappings of resources, models, and trends such as volcanic eruptions, weather patterns, earthquake activity, and mineral deposits. Real estate professionals, architects, and city development agencies use KML to propose construction and visualize plans. Students and teachers use KML to explore people, places, and events, both historic and current. Organizations such as National Geographic, UNESCO, and the Smithsonian have all used KML to display their rich sets of global data.

KML documents and their related images (if any) may be compressed using the ZIP format into KMZ archives. KML documents and KMZ archives may be shared by e-mail, hosted locally for sharing within a private internet, or hosted on a web server.

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, OGC KML 2.3, OGC Keyhole Markup Language 2.3.

iii. Preface

This is an OGC Implementation Standard for encoding representations of geographic data for display in an earth browser.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

1. Galdos Systems Inc.
2. Google Inc.

3. European Union Satellite Centre

v. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Name	Affiliation
David Burggraf	Individual member
Brian McClendon	Google Inc.
Michael Weiss-Malik	Google Inc.
Sean Askay	Google Inc.
Lucio Colaiacomo	European Union Satellite Centre
Richard Martell	Galdos Systems Inc.

1. Scope

KML is an XML grammar used to encode and transport representations of geographic data for display in an earth browser, such as a 3D virtual globe, 2D web browser application, or 2D mobile application. A KML instance is processed in much the same way that HTML (and XML) documents are processed by web browsers. Like HTML, KML has a tag-based structure with names and attributes used for specific display purposes.

KML can be used to:

- Annotate the Earth
- Specify icons and labels to identify locations on the surface of the planet
- Create different camera positions to define unique views for KML features
- Define image overlays to attach to the ground or screen
- Define styles to specify KML feature appearance
- Write HTML descriptions of KML features, including hyperlinks and embedded images
- Organize KML features into hierarchies
- Locate and update retrieved KML documents from local or remote network locations
- Define the location and orientation of textured 3D objects

2. Conformance

This standard defines three conformance classes (levels) for KML resources; these indicate the relative importance or priority of a particular set of constraints. The highest level (CL3) indicates full conformance, but a given application or user community may choose to enforce a lower level of conformance.

1. CL1: Includes test cases covering requirements that must be satisfied in all instances.
2. CL2: As for CL1, plus test cases addressing recommended requirements that should be satisfied.
3. CL3: As for CL2, plus test cases for optional constraints that are essentially informative in nature.

A KML resource that conforms to this standard shall satisfy all requirements stipulated in this document. Furthermore, it shall be checked using all applicable tests specified in Annex A (normative). The framework, concepts, and methodology for testing--and the criteria to be achieved to claim conformance--are specified in the OGC Compliance Testing Program Policies and Procedures manual (OGC 08-134r10) and at the OGC compliance testing web site[1].

3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 14-068, OGC KML 2.3 – Abstract Test Suite

IETF RFC 4287, Atom Syndication Format. Available from: http://tools.ietf.org/html/rfc4287

OASIS Extensible Address Language (XAL) 2.0. Available from: http://www.oasis-open.org/committees/ciq/download.html

IETF RFC 3966, The tel URI for Telephone Numbers. Available from: http://tools.ietf.org/html/rfc3966

ISO 8601:2004, Data elements and interchange formats — Information interchange — Representation of dates and times

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. Available from: http://tools.ietf.org/html/rfc3986

[1] See <http://cite.opengeospatial.org/>.

HTML5. Available at <http://www.w3.org/TR/html5/> W3C Extensible Markup Language (XML) 1.0. Available from: http://www.w3.org/TR/REC-xml/

W3C Namespaces in XML 1.0. Available from: http://www.w3.org/TR/REC-xml-names/

W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. Available at <http://www.w3.org/TR/xmlschema11-1/>

W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. Available at <http://www.w3.org/TR/xmlschema11-2/>

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

4.1
application schema
conceptual schema for data required by one or more applications.

[ISO 19101]

4.2
attribute <XML>
name/value pair contained in an element

NOTE: In this document an attribute is an XML attribute unless otherwise specified

4.3
boundary
set that represents the limit of an entity

[ISO 19107]

4.4
bounding box
minimum volume that encloses a set of objects or data points.

4.5
child element <XML>
immediate descendant element

4.6
complex element <XML>
element of complex content

4.7
coordinate
one of a sequence of n numbers designating the position of a point in n-dimensional space

[ISO 19111]

NOTE: In a coordinate reference system, the n numbers shall be qualified by units.

4.8
coordinate reference system
coordinate system that is related to an object by a datum

[ISO 19111]

4.9
coordinate system
set of mathematical rules for specifying how coordinates are to be assigned to points

[ISO 19111]

4.10
coordinate tuple
tuple composed of a sequence of coordinates

[ISO 19111]

4.11
data type
specification of a value domain with operations allowed on values in this domain

[ISO/TS 19103]

EXAMPLE: integer, real, boolean, string, date (conversion of a data into a series of codes).

NOTE: Data types include primitive predefined types and user-definable types. All instances of a data types lack identity.

4.12
datum
The set of parameters used by a coordinate reference system that define the position of the origin, the scale, and the orientation of a coordinate system

[ISO 19111]

NOTE: The datums in used by the KML coordinate reference system are a vertical datum based on the geoid earth model and a (horizontal) and a geodetic datum, which specifies the ellipsoid model, area of use, and position of the prime meridian .

4.13
document <XML>
well-formed XML instance

4.14
earth browser
software for displaying and annotating models of the Earth

4.15
element <XML>
basic information item of an XML document containing child elements, attributes and character data

NOTE: From the XML Information Set: "Each XML document contains one or more elements, the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by an empty-element tag. Each element has a type, identified by name, sometimes called its 'generic identifier' (GI), and may have a set of attribute specifications. Each attribute specification has a name and a value."

4.16
field
child element of simple content

4.17
field type
XML Schema simple type defined in the KML schema document

4.18
geodetic datum
datum describing the relationship of a 2- or 3-dimensional coordinate system to the Earth

[ISO 19111]

4.19
geographic view
display of geographic KML elements

4.20
interior
set of all points that are on a geometric object but which are not on its boundary

4.21
line string
curve composed of straight-line segments

4.22
list view
display of one or more hierarchies of KML Features

4.23
namespace <XML>
collection of names, identified by a URI reference, which are used in XML documents as element names and attribute names [W3C XML Namespaces]

4.24
plate carrée projection
A simple cylindrical projection in which the target plane has a horizontal axis representing longitude (standard parallel is the Equator) and vertical axis representing latitude. Otherwise known as equi-rectangular, plane chart, or unprojected map projection.

4.25
point
0-dimensional geometric primitive, representing a position

[ISO 19107]

NOTE: The boundary of a point is the empty set.

4.26
polygon
planar surface defined by 1 exterior boundary and 0 or more interior boundaries

4.27
resource
network data object or service that is identified by a URL

4.28
schema
formal description of a model

[ISO 19101]

NOTE: In general, a schema is an abstract representation of an object's characteristics and relationship to other objects. An XML schema represents the relationship between the attributes and elements of an XML object (for example, a document or a portion of a document)

4.29
schema <XML Schema>
collection of schema components within the same target namespace

EXAMPLE: Schema components of W3C XML Schema are types, elements, attributes, groups, etc.

4.30
schema document <XML Schema>
XML document containing schema component definitions and declarations

NOTE: The W3C XML Schema provides an XML interchange format for schema information. A single schema document provides descriptions of components associated with a single XML namespace, but several documents may describe components in the same schema, i.e. the same target namespace.

4.31
simple element <XML>
element of simple content

4.32

tag <XML>

markup in an XML document delimiting the content of an element

NOTE: A tag with no forward slash (e.g. <Placemark>) is called a start-tag (also opening tag), and one with a forward slash (e.g. </Placemark> is called an end-tag (also closing tag).

4.33

tuple

ordered list of values

4.34

Uniform Resource Identifier (URI)

unique identifier for a resource, structured in conformance with IETF RFC 2396

NOTE: The general syntax is <scheme>::<scheme-specific-part>. The hierarchical syntax with a namespace is <scheme>://<authority><path>?<query> – see [RFC 2396].

5. Conventions

5.1 Abbreviated Terms

Some frequently used abbreviated terms:

CRS	Coordinate Reference System
CS	Coordinate System
CSV	Comma Separated Values
CT	Coordinate Transformation
EPSG	European Petroleum Survey Group
GIS	Geographic Information System
GML	Geography Markup Language
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
KMZ	KML Archive File
OGC	Open Geospatial Consortium
RFC	Request for Comments
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
xAL	Extensible Address Language
XML	Extensible Markup Language
XSD	XML Schema Definition
0D	Zero Dimensional
1D	One Dimensional
2D	Two Dimensional

3D	Three Dimensional

5.2 XML Namespaces

Several namespace bindings are assumed in this document (see Table 1). However, in an instance document the actual prefix bound to a given namespace is not significant; any legal NCName (http://www.w3.org/TR/xml-names/#NT-NCName) may be used.

Table 1: Namespace Bindings

Prefix	Namespace name
kml (or no-prefix default)	http://www.opengis.net/kml/2.2
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
atom	http://www.w3.org/2005/Atom
xal	urn:oasis:names:tc:ciq:xsdschema:xAL:2.0

All components of the KML schema are defined in the namespace with the name identifier "`http://www.opengis.net/kml/2.2`", for which the prefix `kml` or the default (no prefix) namespace is used within this Standard. The namespace for KML 2.3 is unchanged from version 2.2, because any KML document valid under the rules of version 2.2 has essentially the same schema assessment semantics under this specification as it did under version 2.2. Users of the KML namespace should be aware that additional backward compatible schema components in this namespace may be defined in future minor revisions of this standard, according to the following rules:

1. New elements or attributes added to previously existing KML elements in a minor revision shall be declared optional.

2. New global elements are permitted in a minor revision, but shall not be required to appear in an instance document.

3. Existing elements/attributes in previous minor versions shall not be removed in a new minor release, but may be labelled 'deprecated', meaning that support may no longer exist in the next major release.

4. Any schema components unrecognized by a KML application shall be gracefully ignored.

The `atom:author` and `atom:link` components described by the Atom Syndication Format are defined in the namespace with the name identifier "`http://www.w3.org/2005/Atom`".

NOTE The schema components from these namespaces are documented in the XML Schema documents that accompany this standard, which are hosted in the OGC KML schema repository: http://schemas.opengis.net/kml/.

The `xAL:AddressDetails` component described by the extensible Address Language is defined in the namespace with the name identifier "`urn:oasis:names:tc:ciq:xsdschema:xAL:2.0`".

5.3 XML Schema

KML 2.3 uses the W3C XML 1.1 Schema language to describe the grammar of conformant KML data instances. The KML schema document is hosted in the OGC KML schema repository (http://schemas.opengis.net/kml/) and forms a normative part of this OGC Implementation Standard.

The normative KML schema includes deprecated schema components that KML producers are advised to not use; they may be removed in a subsequent major revision.

5.4 Versioning

Each schema document specifying components of the KML schema shall carry a version attribute as defined in the XML Schema Recommendation. The format of the version attribute string is x.y.z where x denotes the major version number, y denotes a minor version number, and z denotes a bug fix release for that document. The schema version attribute shall capture the corresponding standard document version and satisfy the compatibility requirements in conformance with OGC standard policy directives (06-135r11, 13.1, 13.2, and 13.4). The target namespace combined with the value of schema/@version uniquely identifies the KML schema.

KML instance documents also carry an version attribute on the root KML element declaring which version of the schema it conforms to. The KML version attribute combined with the KML namespace declared on the root KML element, uniquely identifies the schema that the KML instance conforms to.

5.5 Deprecated parts of previous versions of KML

The verb "deprecate" provides notice that the referenced portion of this Standard is being retained for backwards compatibility with earlier versions but may be removed from a subsequent major revision.

5.6 Documentation

Clauses 7 to 16 specify the content model for each KML element. The KML content model is defined in terms of child elements and attributes. Where a child element is defined in a separate subclause, a link is provided to it.

Throughout this document, the following conventions are used:

- Concrete element and attribute names used within the text are formatted in `Courier New` and abstract elements in *`Courier New Italic`*, except in the case of headings and references to them.
- A value from an enumerated set is in **bold**.
- Child element content is specified within the element subclause whose type declares such children.
- The prefix "xsd:" is used to identify types that are defined by XML Schema, for example xsd:string.
- The use of abstract substitution elements within the text refers to the concrete elements that substitute for them. For example, *`kml:AbstractContainerGroup`* refers to the `kml:Document` and `kml:Folder` elements that substitute for it. Element substitution is described for each element where applicable.
- The term `kml:Document` is a KML element; the term "KML document" refers to an XML instance of KML.
- Default values for simple elements are listed under relevant Content subclauses. Unless otherwise stated, default values for optional elements shall apply when such elements are empty or absent.
- The term "KML resource" refers to a local or remote KML or KMZ instance.

Each element subclause includes an XML structure section that shows the content model for the described element as a representative XML instance, using the following conventions:

- The minimum and maximum occurrence of elements and attributes required by its schema type are shown in square brackets, for example [0..1]. Such occurrence requirements are necessary but insufficient in that additional occurrence requirements are asserted within this .
- Elements are itemized in document order but may appear in any order in an XML instance, since element order is no longer enforced by the KML schema as of version 2.3.
- Where the type for an element derives by extension from a base type, the elements and attributes that have been added to the base type's content are shown in **bold**.

6. KML Model Overview

6.1 KML Abstract Element Substitution Hierarchy

This section provides an overview of the KML schema type hierarchy. While element names are used, they represent the schema types that define their content. As KML is an XML grammar, element names are case-sensitive and must appear exactly as specified in the KML schema document.

In KML, some types are derived from a parent type. A derived type typically inherits all of the elements of its parent type and adds some specific element content of its own. KML also includes abstract elements whose type is also abstract. Such abstract types are used to establish schema type hierarchies. Abstract elements may serve as placeholders for elements that substitute for them in the XML Schema sense.

The KML abstract element substitution hierarchy is represented by the following nested list, showing the abstract KML elements that serve as the head of substitution groups. The corresponding types of the abstract elements (shown in brackets) follow an equivalent parallel type hierarchy. For example, *kml:AbstractContainerGroup* substitutes for *kml:AbstractFeatureGroup*, which in turn substitutes for *kml:AbstractObjectGroup*. Likewise, kml:AbstractContainerType derives by exenstion from kml:AbstractFeatureType, which in turn derives from kml:AbstractObjectType.

- *kml:AbstractObjectGroup (kml:AbstractObjectType)*
    - *kml:AbstractFeatureGroup (kml:AbstractFeatureType)*
        - *AbstractContainerGroup (kml:AbstractContainerType)*
        - *AbstractOverlayGroup (kml:AbstractOverlayType)*
    - *AbstractGeometryGroup (kml:AbstractGeometryType)*
    - *AbstractStyleSelectorGroup (kml:AbstractStyleSelectorType)*
    - *AbstractSubStyleGroup (kml:AbstractSubStyleType)*
        - *AbstractColorStyleGroup (kml:AbstractColorStyleType)*
    - *AbstractExtentGroup (kml:AbstractExtentType)*
        - *AbstractLatLonBox (kml:AbstractLatLonBoxType)*
    - *AbstractViewGroup (kml:AbstractViewType)*
    - *AbstractTimePrimitiveGroup (kml:AbstractTimePrimitiveType)*
    - *AbstractLinkGroup (kml:AbstractObjectType)*
    - *AbstractTourPrimitiveGroup (kml:AbstractTourPrimitiveType)*

These abstract elements and their associated base types define the core of the language.

All concrete elements derived from *kml:AbstractObjectGroup* may have an assigned identifier.

Simple elements are generally referred to as KML fields. Such elements are of XML Schema simple types or KML field types defined in the KML schema. KML field types are specified in clause 16 Field Types.

6.2 Coordinate Reference System

Each element that extends the `kml:AbstractGeometryGroup` element defines a spatial extent of a `kml:Placemark`. The spatial extent may include the location of an anchor point on the earth to serve as an origin for a 3D object as in the case of the `kml:Model` element, or may include the encoding of explicit coordinate tuples in the `kml:coordinates` element in the case of the `kml:Point`, `kml:LineString`, and `kml:LinearRing` elements.

The KML encoding of every `kml:Location` and coordinate tuple uses geodetic longitude, geodetic latitude, and altitude (in that order) as defined in Annex B. Note that altitude is measured from the vertical datum, which is the WGS84 EGM96 Geoid. The altitude measurement (orthometric *H*) is illustrated in Figure 1.

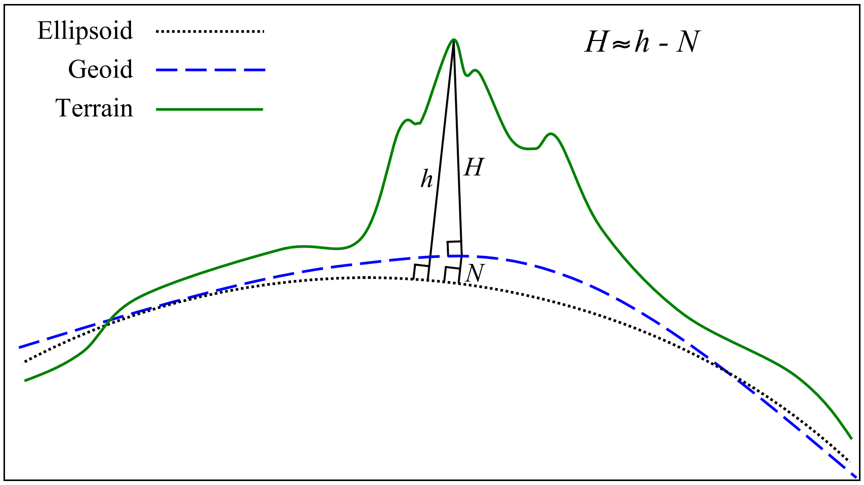


Figure 1: Altitude H is Measured from the Vertical Datum (Geoid) and is Compared to the Ellipsoid Height h and the Geoid Undulation N.

6.3 Geometry Interpolation for 3D Earth Browsers

6.3.1 Interpolated Points

Geometric points which are not explicitly encoded are called interpolated points. The following subclauses describe the interpolation schemes for the `kml:LineString`, `kml:LinearRing`, and `kml:Polygon` elements.

6.3.2 kml:LineString and kml:LinearRing

The type of interpolation used for the `kml:LineString` and `kml:LinearRing` elements depend on the values of the child `kml:altitudeMode` and `kml:tessellate` elements. If the `kml:altitudeMode` value is not **clampToGround** then the interpolation between two consecutive control points is a straight line segment in the 3D WGS 84 geocentric coordinate reference system (http://www.opengis.net/def/crs/EPSG/0/4978). This straight line segment will be referred to as *L* in Table 2, which summarizes the `kml:LineString` and `kml:LinearRing` interpolation scheme for the various combinations of `kml:altitudeMode` and `kml:tessellate` values.

Table 2: Interpolation scheme for `kml:LineString` and `kml:LinearRing`

<altitudeMode>	***<tessellate>***	***Interpolation between control points***
relativeToGround or absolute	0 (false) or 1 (true)	A straight line segment *L* in the 3D WGS 84 geocentric coordinate reference system (http://www.opengis.net/def/crs/EPSG/0/4978)
clampToGround	1 (true)	Project each point of *L* to the terrain surface along a line through the earth's center of mass
clampToGround	0 (false)	First project each control point to the terrain surface along a line through the earth's center of mass, then interpolate between the projected control points along a straight line segment in the 3D WGS 84 geocentric coordinate reference system (http://www.opengis.net/def/crs/EPSG/0/4978)

For example, in the following KML instance the `kml:LineString` coordinates element has two control points (–135,30,500000) and (–80,30,500000) of the form (lon,lat,altitude) in the CRS defined in with `gml:id="LonLat84_5773"`.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <LineString>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>-135,30,500000 -80,30,500000</coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Since the `kml:altitudeMode` is **absolute**, the interpolation between these two control points is the straight line segment *L* shown in Figure 2 in the 3D WGS 84 geocentric CRS, which does not follow the earth's curvature and cuts through the earth's terrain.

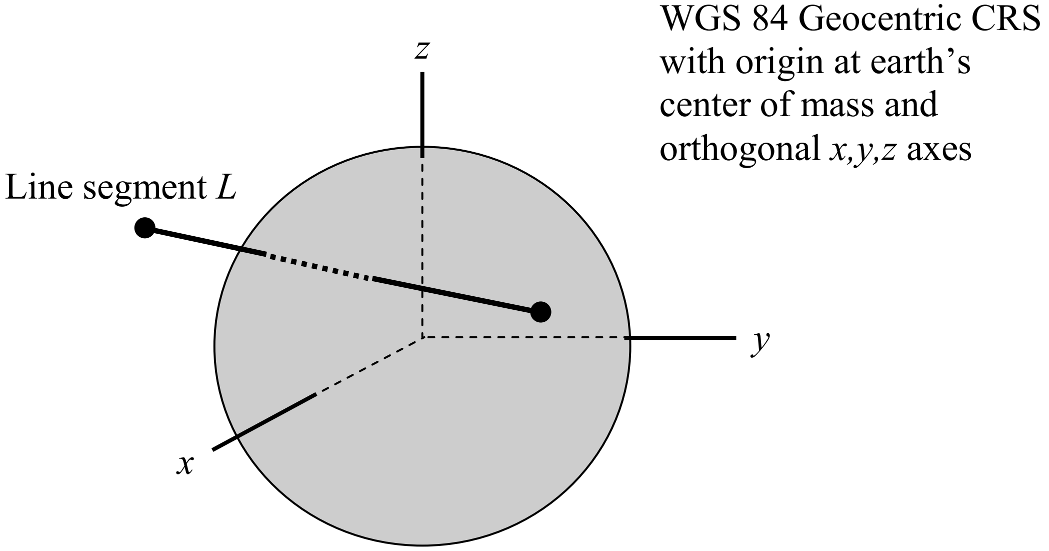


Figure 2: `kml:LineString` **Interpolation with** `altitudeMode` **= ‘**`relativeToGround`**’ or '**`absolute`**' and any value of** `tessellate`

The line segment *L* will be projected to the terrain surface if `kml:altitudeMode` and `kml:tessellate` are set as in the following `kml:LineString` instance. In this case the projected `kml:LineString` will follow the earth's curvature as shown in Figure 3

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <LineString>
        <tessellate>true</tessellate>
        <altitudeMode>clampToGround</altitudeMode>
        <coordinates>-135,30,500000 -80,30,500000</coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

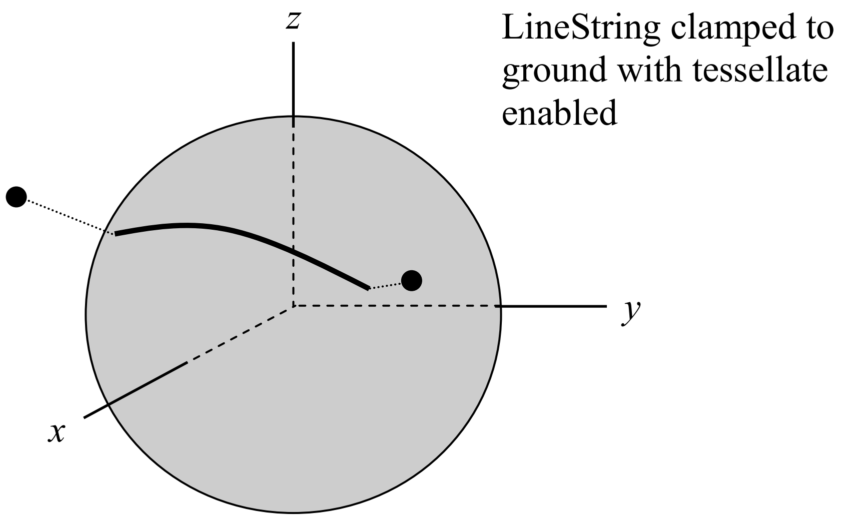


Figure 3: `kml:LineString` **Interpolation with** `altitudeMode` **= ‘`clampToGround`’ and** **`tessellate` = ‘`true`’**

If `kml:altitudeMode` and `kml:tessellate` are set as in the following `kml:LineString` instance, then the only the control points are projected to the terrain and the interpolation between the projected control points is a straight line segment in the WGS 84 Geocentric CRS as shown in Figure 4.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <LineString>
        <tessellate>false</tessellate>
        <altitudeMode>clampToGround</altitudeMode>
        <coordinates>-135,30,500000 -80,30,500000</coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

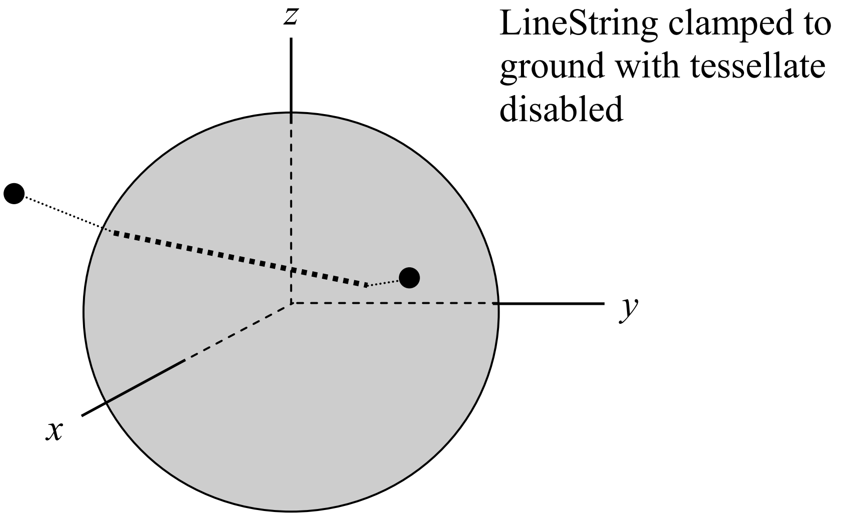


Figure 4: `kml:LineString` **Interpolation with** `altitudeMode` **= '**`clampToGround`**' and** `tessellate` **= '**`false`**'**

6.3.3 kml:Polygon

The type of interpolation used for the `kml:Polygon` element also depends on the values of the child `kml:altitudeMode` and `kml:tessellate` elements. If the `kml:altitudeMode` value is not **clampToGround** then the interpolation of the `kml:Polygon` boundary comprised of the descendent `kml:LinearRing` elements is as described previously in Table 2. The remaining interior points of the `kml:Polygon` are then filled in linearly in the 3D WGS 84 geocentric CRS, i.e. they must lie on the plane that passes through all the control points of each `kml:LinearRing`.

NOTE: The control points of every `kml:LinearRing` must lie on a common plane.

Table 3 summarizes the `kml:Polygon` interpolation scheme for the various combination of `kml:altitudeMode` and `kml:tessellate` values.

Table 3: Interpolation scheme for kml:Polygon

<altitudeMode>	***<tessellate>***	***Interpolation between control points***
relativeToGround or absolute	0 (false) or 1 (true)	Boundary points of the kml:Polygon in the descendent kml:LinearRing(s) are interpolated as in Table 2 and the interior points are filled in linearly in the 3D WGS 84 geocentric coordinate reference system (http://www.opengis.net/def/crs/EPSG/0/4978), i.e. they must lie on a plane.
clampToGround	0 (false) or 1 (true)	The boundary control points of each descendent kml:LinearRing are first projected to the plate carrée plane (where altitude is dropped), then straight line segment interpolation in the plate carrée (lon,lat) plane is used between consecutive control points. The interior points are then filled in linearly in the plate carrée plane. Finally, the (lon,lat) points of the polygon in the plate carrée plane are mapped back to (lon.lat,alt) points on the earth's terrain surface model

For example, the following `kml:Polygon` encodes five control points in its outer boundary in the CRS defined in Annex B.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <Polygon>
        <altitudeMode>absolute</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
-135,50,300000 -135,40,450000 -80,40,450000 -80,50,300000 -135,50,300000
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```

Since the `kml:altitudeMode` is `absolute`, the outer boundary points of the polygon that are interpolated between the control points in the `kml:LinearRing` form a quadrilateral perimeter in the 3D WGS 84 geocentric CRS. The interior points of this 4 sided polygon are filled in linearly in the 3D WGS 84 geocentric CRS and form the plane region inside the perimeter. Note that the plane region does not follow the earth's curvature and cuts through the surface of the earth as shown in Figure 5.

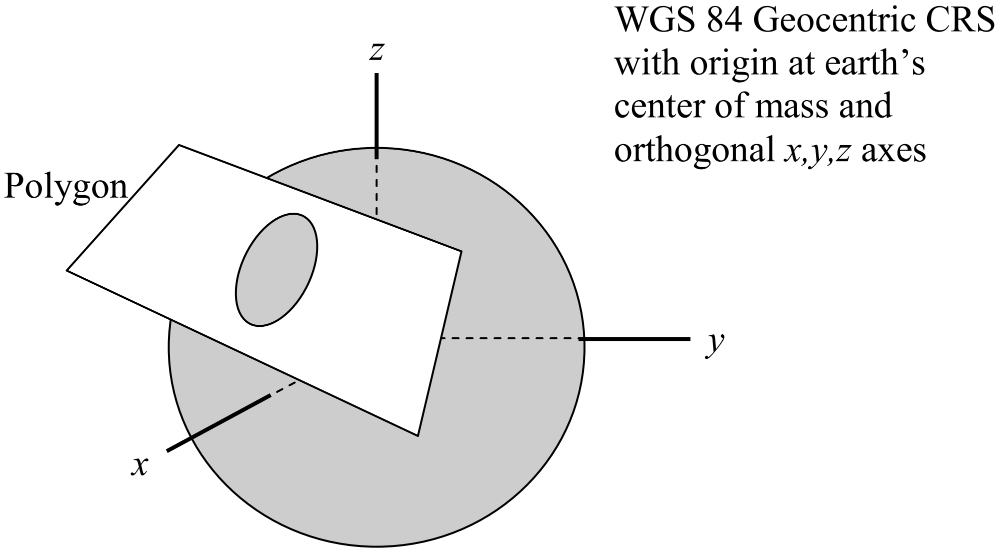
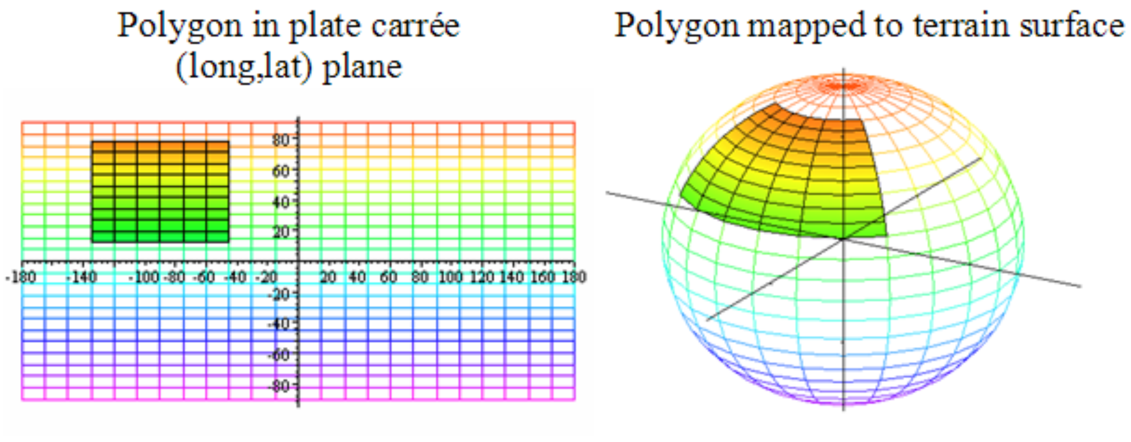


Figure 5: kml:Polygon Interpolation with `altitudeMode` = '`relativeToGround`' or '`absolute`' and any value of `tessellate`

In the following `kml:Polygon` instance the `kml:altitudeMode` is set to **clampToGround**.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <Polygon>
        <altitudeMode>clampToGround</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>-135,78.5,300000 -135,12.5,300000 -45,12.5,300000 -
45,78.5,300000 -135,78.5,300000</coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```

The outer boundary points of the polygon that are interpolated between the control points in the `kml:LinearRing` form a rectangular perimeter in the plate carrée plane. Then the interior points of the rectangle are filled in linearly in this plane. Finally each (lon,lat) point of the rectangle is mapped to a (lon,lat,alt) point on the earth terrain surface as shown in Figure 6.


Polygon in plate carrée (long,lat) plane

Polygon mapped to terrain surface

Figure 6: kml:Polygon Interpolation with altitudeMode = 'clampToGround' and any value of tessellate

6.3.4 kml:GroundOverlay and kml:Region

The latitude and longitude boundaries for both `kml:GroundOverlay` and `kml:Region` are specified by the elements `kml:west`, `kml:east`, `kml:south`, and `kml:north`. In the case of `kml:GroundOverlay` constant lines of longitude demarcate the portion of the `kml:LatLonBox` boundary corresponding to the values of `kml:west`, `kml:east` and constant lines of latitude demarcate the portion of the boundary corresponding to the values of `kml:south`, `kml:north`. If `kml:altitudeMode` value is not `clampToGround`, as in the following KML example, then the altitude of the `kml:GroundOverlay` is determined by the `kml:altitude` element.

```
<kml>
  <GroundOverlay>
    <altitude>500000</altitude>
    <altitudeMode>absolute</altitudeMode>
    <LatLonBox>
      <north>80.0</north>
      <south>60.0</south>
      <east>60.0</east>
      <west>-60.0</west>
    </LatLonBox>
  </GroundOverlay>
</kml>
```

The KML example above is shown in Figure 7. If `kml:altitudeMode` value is **clampToGround** then the `kml:GroundOverlay` is projected onto the earth's terrain surface model similar to the case of the `kml:Polygon` shown on the right hand side of Figure 6.

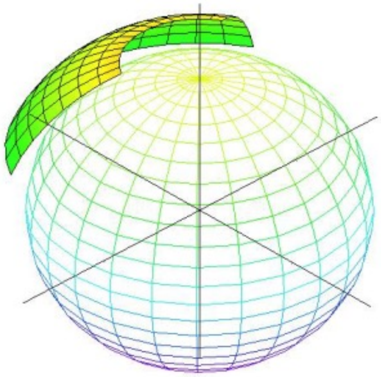


Figure 7: KML GroundOverlay when value of `kml:altitudeMode` is not clampToGround

In the case of `kml:Region` the values of `kml:west`, `kml:east` of `kml:LatLonAltBox` determine the boundary surfaces of constant longitude and the values of `kml:south`, `kml:north` determine the boundary surfaces of constant latitude. The values of `kml:minAltitude` and `kml:maxAltitude` in the `kml:LatLonAltBox` determine the lower and upper bounding surfaces of constant altitude, respectively.

```
<kml>
  <Region>
    <LatLonAltBox>
      <north>80.0</north>
      <south>60.0</south>
      <east>60.0</east>
      <west>-60.0</west>
      <minAltitude>500000</minAltitude>
      <maxAltitude>2000000</maxAltitude>
      <altitudeMode>absolute</altitudeMode>
    </LatLonAltBox>
  </Region>
</kml>
```

The spatial extent of the sample `kml:Region` encoded above is depicted in Figure 8.

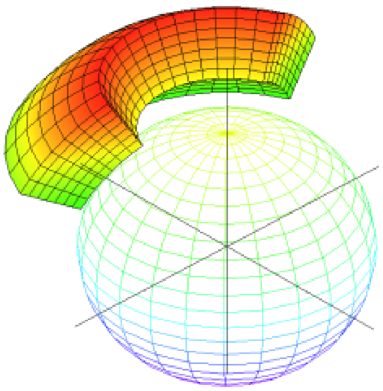


Figure 8: KML Region when value of `kml:altitudeMode` is not clampToGround

6.4 Shared Styles

A `kml:Style` or `kml:StyleMap` element contained by a *`kml:AbstractFeatureGroup`* element is an "inline style" and shall apply only to the *`kml:AbstractFeatureGroup`* that contains it. When these elements are encoded as the child of a `kml:Document` element they are called a "shared style." A shared style shall have an `id` value. A shared style applies to any *`kml:AbstractFeatureGroup`* that references the style from its child `kml:styleUrl` element.

If a *`kml:AbstractFeatureGroup`* is associated with both an inline and shared style, the inline style shall take precedence.

Shared styles shall only be encoded within a `Document`. Shared styles are not inherited by any child *`kml:AbstractFeatureGroup`* elements of a `kml:Document`.

For a `kml:Style` or `kml:StyleMap` that applies to a `kml:Document`, the `kml:Document` itself must explicitly reference a shared style. For example:

```
<Document>
  <Style id="myPrettyDocument">
   <ListStyle>...</ListStyle>

  </Style>
  <styleUrl#myPrettyDocument">
  ...
</Document>
```

The following example illustrates the use of a shared style.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>Document.kml</name>
  <open>1</open>
  <Style id="exampleStyleDocument">
    <LabelStyle>
      <color>ff0000cc</color>
    </LabelStyle>
  </Style>
  <Placemark>
    <name>Document Feature 1</name>
    <styleUrl>#exampleStyleDocument</styleUrl>
    <Point>
      <coordinates>-122.371,37.816,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>Document Feature 2</name>
    <styleUrl>#exampleStyleDocument</styleUrl>
    <Point>
      <coordinates>-122.370,37.817,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

6.5 Entity Replacement

Entity substitution is employed as a template mechanism within the `kml:BalloonStyle` `kml:text` element. Individual values shall be substituted for each instance of the entity, or a null string if no value exists. The source of values for entity substitution is local to the *`kml:AbstractFeatureGroup`* being styled and any `kml:Schema` elements associated with it. Entity syntax for identifying a substitution value is as follows:

1. **$[element_or_attribute_name]**, where "element_or_attribute_name" is the name of a field element or attribute of the *`kml:AbstractFeatureGroup`*. This identifies the value of the field element or attribute.
2. **$[name_attribute_of_Data_element]**, where "name_attribute_of_Data_element" is the value of the `kml:name` attribute of a descendant `kml:Data` element of the *`kml:AbstractFeatureGroup`*. This identifies the value of the child `kml:value` element of the `kml:Data` element.
3. **$[name_attribute_of_Data_element/displayName]**, where "name_attribute_of_Data_element" is the value of the `kml:name` attribute of a

descendant `kml:Data` element of the *`kml:AbstractFeatureGroup`*; "/" is a separator; and "displayName" is the value of the kml:Data/kml:displayName element.

4. **$[TYPENAME/TYPEFIELD]**, where "TYPENAME" is the value of the `kml:name` attribute of a descendant `kml:Schema` element of the *`kml:AbstractFeatureGroup`*; "/" is a separator; and "TYPEFIELD" is the value of the `kml:name` attribute of a child `kml:SimpleField` element of the `kml:Schema` element . This identifies the value of a descendant `kml:SimpleData` element of the *`kml:AbstractFeatureGroup`* that references the `kml:SimpleField` element.
5. **$[TYPENAME/TYPEFIELD/displayName]**, where "TYPENAME" is the value of the `kml:name` attribute of a descendant `kml:Schema` element of the *`kml:AbstractFeatureGroup`*; "/" is a separator; and "TYPEFIELD" is the value of the `kml:name` attribute of a child `kml:SimpleField` element of the `kml:Schema` element; and "displayName" is the value of the child `kml:displayName` element of the `kml:SimpleField` element. This identifies the value of the `kml:displayName` element.

For example, the **$[name]** and **$[description]** entities in the following `kml:BalloonStyle kml:text` element shall be replaced by the `kml:name` and `kml:description` values of *`kml:AbstractFeatureGroup`* elements associated with the `kml:BalloonStyle`:

```
<text>This is $[name], whose description is:<br/>$[description]</text>
```

6.6 Application Profiles

6.6.1 Introduction

An application profile defines a set of elements derived from one or more base standards for the purpose of:

- promoting interoperability; and
- meeting the requirements of a particular application domain.

6.6.2 Rules for Authoring KML Application Profiles

A KML profile can restrict or extend KML to the extent permitted by this standard, see 6.7 (Extension Model). The KML schema provides a number of extension points that may be exploited in a profile. While a profile shall not contradict the standard, it may restrict the choice of options or introduce new elements (or do both).

Documents that conform to the KML standard may contain elements and attributes that are not part of the standard but are defined in an application profile. Such elements and attributes—called foreign information items—must not reside within the KML namespace; they shall be placed in another namespace.

Application profiles shall not redefine any KML components (structurally or semantically) within the KML namespace.

Application profiles which extend KML shall:

- define the XML structure of any new elements and attributes in an application profile document. If the application profile requires the use of newly created schema components (not including elements and attributes that are reused from a third party schema), the newly created components shall be defined in a valid application profile schema that imports the KML schema;
- add new elements and/or attributes to existing concrete KML elements according to 6.7.2 Extension by Composition, where permitted by the KML schema;
- derive any new complex types of complex content in the application profile schema by extension according to 6.7.1Extension by Inheritance; from the relevant KML abstract type whose semantics it shares. If the new complex type does not share semantics with any existing KML abstract type, then derive by extension from kml:AbstractObjectType.

  EXAMPLE: a new feature type shall derive from kml:AbstractFeatureType.

- declare any new elements and attributes in the application profile schema as optional, i.e. minOccurs="0", to support the KML update mechanism. This however does not preclude asserting minimum occurrence constraints as conformance rules within supplementary normative application profile documentation.

Application profiles that extend KML should:

- place any newly created extension elements and attributes in a "vendor-neutral" namespace to support any future potential integration with the KML standard, and encourage interoperability in general.

An adopted OGC application profile that extends KML should:

- be based on a source mass market application profile that is supported by running code;
- preferably maintain or else deprecate the namespace of a source application profile from which it derives, if/where the source profile is in popular use within the mass market. This is to ensure backwards compatibility with existing instances and consumers of the source profile;
- provide a structural and semantic mapping between any profile components which have changed from their original source profile.

Authors of application profiles are encouraged to submit their extensions to OGC for standardization. This does not preclude any resulting OGC application profile from being merged later into the KML standard.

6.7 Extension Model

The KML schema provides several mechanisms to extend KML within application profiles. All KML extensions shall conform to the application extension profile requirements discussed in 6.6 Application Profiles.

6.7.1 Extension by Inheritance

The KML schema defines abstract base types (e.g. *`kml:AbstractFeatureType`*), concrete types (e.g. `kml:PlacemarkType`), and abstract elements that act as the head of substitution groups (e.g. *`kml:AbstractFeatureGroup`*). While further derivation of all KML concrete complex types is prohibited, new schema types in a KML Application Profile may be derived from the core abstract base types, listed in 6.1.

6.7.2 Extension by Composition

6.7.2.1 Simple Element Substitution

In a KML Application Profile, a simple element (whose type derives from `xsd:anySimpleType`) may be declared to substitute for a KML abstract head element that is:

- of type `xsd:anySimpleType`. Such abstract head elements have a naming convention whereby their local name ends with "SimpleExtensionGroup" (e.g. *`kml:AbstractFeatureSimpleExtensionGroup`*).
- of type `kml:enumBaseType`. Such abstract head elements have a naming convention whereby the local name begins with "abstract" (e.g. *`kml:abstractFlyToMode`*).

Usage examples:

- a new element whose type validly derives from *`xsd:anySimpleType`* and substitutes for *`kml:AbstractFeatureSimpleExtensionGroup`* would be available wherever *`kml:AbstractFeatureSimpleExtensionGroup`* is permitted; that is, as a child of `kml:Document`, `kml:Folder`, `kml:GroundOverlay`, `kml:ScreenOverlay`, `kml:PhotoOverlay`, `kml:Placemark`, and `kml:NetworkLink`.
- a new element whose type defines new enumeration values by deriving by restriction from `kml:enumBaseType` and substitutes for `kml:abstractFlyToMode` would be available wherever `kml:abstractFlyToMode` is permitted; that is, as a child of `kml:FlyTo`.

6.7.2.2 Complex Element Substitution

In a KML Application Profile, a complex element may be declared to substitute for a KML abstract head element that is:

- of a core abstract base type (e.g. *`kml:AbstractFeatureType`*).

- of type `xsd:anyType` and substitutes for *`kml:AbstractObjectGroup`*. Substituting for such an element requires a type definition that derives from *`kml:AbstractObjectGroup`*. All such head elements have a local name that ends with "ObjectExtensionGroup" (e.g. *`kml:AbstractFeatureObjectExtensionGroup`*).
- of type `xsd:anyType` and declares no substitution group. These have a local name that ends with "Extension" (e.g. *`kml:DataExtension`*).

Usage examples:

- a new element whose type derives by extension from *`kml:AbstractFeatureType`* and substitutes for *`kml:AbstractFeatureGroup`* would be available wherever *`kml:AbstractFeatureGroup`* is permitted; that is, as a child of `kml:Document`, `kml:Folder`, `kml:kml` or `kml:Delete`.
- a new element whose type derives by extension from *`kml:AbstractObjectType`* and substitutes for *`kml:AbstractFeatureObjectExtensionGroup`* would be available wherever *`kml:AbstractFeatureObjectExtensionGroup`* is permitted; that is, as a child of `kml:Document`, `kml:Folder`, `kml:GroundOverlay`, `kml:ScreenOverlay`, `kml:PhotoOverlay`, `kml:Placemark`, and `kml:NetworkLink`.
- a new element whose type is of `xsd:anyType` and substitutes for *`kml:DataExtension`* would be available wherever *`kml:DataExtension`* is permitted; that is, as a child of `kml:Data`.

An example of the creation of new extension elements using complex element substitution is provided in Annex D.1 Extension Model Examples.

6.7.2.3 Foreign Wildcard Elements

KML 2.3 makes use of the new `defaultOpenContent` element introduced in XML Schema 1.1 to optionally allow for any element content from a foreign namespace. This extension mechanism allows for the direct use of XML element content from third-party schemas. Authors of KML Application Profile extensions are now able to experimentally add foreign element content interleaved among existing KML elements at any location in the KML document tree. Further details and examples of this type of extension-by-composition as applied to KML style element extensions are illustrated in Annex D.2 Extension Model Examples.

6.7.2.4 Foreign Wildcard attributes

KML 2.3 also introduces a new extension point for attributes. Any number of 'wildcard attributes' from a foreign namespace, which can be utilized on most KML elements, including all elements whose type ultimately derives from any of the types listed in 6.7.2.4.1. The so-called 'wildcard attributes' are realized in XML Schema as an `xsd:anyAttribute`. declaration as follows:

```
    <anyAttribute namespace="##other" processContents="lax"/>
```

By setting `processContents` to 'lax', wildcard content can be validated if a schema location is supplied in the instance document (i.e. using `xsi:schemaLocation`); if the schema location is not supplied then validation of the wildcard attributes will not occur. The `xsd:anyAttribute` is included by reference from all of the types listed in 6.7.2.4.1. An example of the creation of a new extension attribute is provided in Annex D.3 Extension Model Examples.

6.7.2.4.1 KML types declared with the anyAttribute wildcard attribute

- kml:AbstractObjectType,
- kml:vec2Type,
- kml:ExtendedDataType,
- kml:SchemaDataType,
- kml:SimpleDataType,
- kml:SimpleArrayDataType,
- kml:DataType,
- kml:KMLType,
- kml:NetworkLinkControlType,
- kml:SchemaType,
- kml:SimpleFieldType,
- kml:SimpleArrayFieldType,
- kml:BoundaryType,
- kml:UpdateType,
- kml:CreateType,
- kml:DeleteType,
- kml:ChangeType.

7. Root Element

7.1 kml

7.1.1 Structure

```
<kml:kml
 hint="string [0..1]"
 version="kml:kmlVersionType [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:NetworkLinkControl>...</kml:NetworkLinkControl> [0..1]
    <kml:AbstractFeatureGroup>...</kml:AbstractFeatureGroup> [0..1]
    <kml:KmlSimpleExtensionGroup>...</kml:KmlSimpleExtensionGroup> [0..*]
    <kml:KmlObjectExtensionGroup>...</kml:KmlObjectExtensionGroup> [0..*]
</kml:kml>
```

7.1.2 Description

The root element of a KML document instance. It should contain at least one of its child elements.

7.1.3 Content

7.1.3.1 kml:NetworkLinkControl

See 13.4 kml:NetworkLinkControl.

7.1.3.2 kml:AbstractFeatureGroup

See 9.1 kml:AbstractFeatureGroup.

7.1.3.3 kml:KmlSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

7.1.3.4 kml:KmlObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

7.1.4 Attributes

7.1.4.1 hint

7.1.4.1.1 Description

The `hint` attribute may be used to provide information on how to process the KML document instance.

7.1.4.1.2 Content

Type:	xsd:string
Default Value:	none

7.1.4.2 version

7.1.4.2.1 Description

The `version` attribute declares which version of the schema the KML instance conforms to.

7.1.4.2.2 Content

Type:	kml:kmlVersionType
Default Value:	2.2.0

7.1.4.3 anyAttribute

7.1.4.3.1 Description

The `anyAttribute` declaration provides a new extension point in KML 2.3, which allows any number of ‘wildcard attributes’ from a foreign namespace (see 6.7.2.4 Foreign Wildcard attributes).

7.1.4.3.2 Content

Type:	xsd:anySimpleType
Default Value:	none

7.1.5 Assertions

If any of the following descendent elements (or element/attribute combinations) are present then the attribute version value shall be '2.3.0' or later:

- `kml:altitudeOffset`
- `kml:balloonVisibility`
- `kml:Camera/kml:TimeStamp`
- `kml:Camera/kml:TimeSpan`
- `kml:Camera/kml:horizFov`
- `kml:Create/kml:MultiGeometry`
- `kml:Data/@uom`
- `kml:Delete/kml:MultiGeometry`
- `kml:Delete/kml:Point`
- `kml:Delete/kml:LineString`

- kml:Delete/kml:LinearRing
- kml:Delete/kml:Polygon
- kml:Delete/kml:Model
- kml:LatLonQuad
- kml:LookAt/kml:TimeStamp
- kml:LookAt/kml:TimeSpan
- kml:LookAt/kml:horizFov
- kml:MultiTrack
- kml:seaFloorAltitudeMode
- kml:SimpleArrayField
- kml:SimpleField/@uom
- kml:Tour
- kml:Track

7.1.6 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2" version="2.3.0">
  <NetworkLinkControl>...</NetworkLinkControl>
  <Document>...</Document>
</kml>
```

8. Object

8.1 kml:AbstractObjectGroup

8.1.1 Structure

```
<kml:AbstractObjectGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
</kml:AbstractObjectGroup>
```

8.1.2 Description

The following elements can be used wherever this abstract element is referenced:

- *kml:AbstractExtentGroup*
- *kml:AbstractFeatureGroup*
- *kml:AbstractGeometryGroup*
- *kml:AbstractLinkGroup*
- *kml:AbstractStyleSelectorGroup*
- *kml:AbstractSubStyleGroup*
- *kml:AbstractTimePrimitiveGroup*
- *kml:AbstractTourPrimitiveGroup*
- *kml:AbstractViewGroup*
- *kml:Alias*
- *kml:Data*
- *kml:Icon*
- *kml:ImagePyramid*
- *kml:ItemIcon*
- *kml:Location*
- *kml:Lod*
- *kml:Orientation*
- *kml:Pair*
- *kml:Playlist*
- *kml:Region*
- *kml:ResourceMap*
- *kml:Scale*
- *kml:SchemaData*
- *kml:ViewVolume*

8.1.3 Content

8.1.3.1 kml:ObjectSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

8.1.4 Attributes

8.1.4.1 id

8.1.4.1.1 Description

The `id` attribute may be used to specify a unique identifier for the *kml:AbstractObjectGroup* within the KML document instance.

If an element affiliated with *`kml:AbstractObjectGroup`* is not being updated (that is, it is not a descendant of `kml:Update`) and it is empty then it shall have an `id` attribute; otherwise it cannot be updated. An object that is not empty should have an identifier to permit future updates.

8.1.4.1.2 Content

Type:	xsd:ID
Default Value:	none

8.1.4.2 targetId

8.1.4.2.1 Description

The optional `targetId` attribute may be used to encode the id value of another *`kml:AbstractObjectGroup`*.

If a *`kml:AbstractObjectGroup`* is being used for update purposes (is a grandchild of `kml:Update)` then it shall have a `targetId` attribute referencing the *`kml:AbstractObjectGroup`* element to be updated. Otherwise, outside of an update context `targetId` has no meaning.

See also 13.5 kml:Update.

8.1.4.2.2 Content

Type:	xsd:NCName
Default Value:	none

8.1.4.3 anyAttribute

See 7.1.4.3 anyAttribute.

9. Features

9.1 kml:AbstractFeatureGroup

9.1.1 Structure

```
<kml:AbstractFeatureGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml:AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
</kml:AbstractFeatureGroup>
```

9.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- *kml:AbstractContainerGroup*
- *kml:AbstractOverlayGroup*
- kml:NetworkLink
- kml:Placemark
- kml:Tour

9.1.3 Content

9.1.3.1 kml:name

9.1.3.1.1 Description

Specifies a label for the *kml:AbstractFeatureGroup*.

9.1.3.1.2 Content

Type:	xsd:string
Default Value:	none

9.1.3.2 kml:visibility

9.1.3.2.1 Description

Specifies whether the *kml:AbstractFeatureGroup* shall be drawn in the geographic view when it is initially loaded (1 or true), or not (0 or false). In order for a *kml:AbstractFeatureGroup* to be visible, the kml:visibility tag of all its ancestors shall also be set to 1 or true.

9.1.3.2.2 Content

Type:	xsd:boolean
Default Value:	1 or true

9.1.3.3 kml:balloonVisibility

9.1.3.3.1 Description

Specifies whether the description balloon is opened when loaded or updated.

9.1.3.3.2 Content

Type:	xsd:boolean
Default Value:	1 or true

9.1.3.4 kml:open

9.1.3.4.1 Description

Specifies whether a *kml:AbstractContainerGroup* appears expanded (1 or true) or collapsed (0 or false) when first loaded into the list view.

See also 12.17 kml:ListStyle.

9.1.3.4.2 Content

Type:	xsd:boolean
Default Value:	0 or false

9.1.3.5 atom:author

9.1.3.5.1 Description

Specifies the author of the *kml:AbstractFeatureGroup*. See also 9.8 kml:AbstractContainerGroup regarding the inheritance of atom:author within KML feature hierarchies.

9.1.3.5.2 Content

See IETF RFC 4287, 4.2.1.

9.1.3.6 atom:link

9.1.3.6.1 Description

Specifies the URL of the source resource that contains the *kml:AbstractFeatureGroup*. The URL is encoded as the value of the atom:href attribute.

The atom:link rel attribute shall be present and its value should be **related**. See also 9.8 kml:AbstractContainerGroup regarding the inheritance of the atom:link within KML feature hierarchies.

9.1.3.6.2 Content

See IETF RFC 4287, 4.2.7.

9.1.3.7 kml:address

9.1.3.7.1 Description

A string value representing an unstructured address for the *kml:AbstractFeatureGroup* such as street, city, state address, and/or a postal code. This may be used to geocode the location of a *kml:AbstractFeatureGroup* if it does not contain a *kml:AbstractGeometryGroup* element.

9.1.3.7.2 Content

Type:	xsd:string
Default Value:	none

9.1.3.8 xal:AddressDetails

9.1.3.8.1 Description

A structured address for the *kml:AbstractFeatureGroup* formatted according to xAL 2.0. This may be used to geocode the location of a *kml:AbstractFeatureGroup* if it does not contain a *kml:AbstractGeometryGroup* element.

9.1.3.8.2 Content

See OASIS Extensible Address Language (XAL) 2.0.

9.1.3.9 kml:phoneNumber

9.1.3.9.1 Description

A value representing a telephone number. The number should be formatted according to IETF RFC 3966.

9.1.3.9.2 Content

Type:	xsd:string
Default Value:	none

9.1.3.10 kml:snippet

9.1.3.10.1 Description

Specifies a short description of the *`kml:AbstractFeatureGroup`*. The value of `kml:snippet`, if present, is used in the list view instead of `kml:description`.

The text may include HTML content that is encoded as well-formed XML using HTML entity references or by enclosing the HTML within a CDATA section.

The element `kml:snippet` replaces the deprecated element `kml:Snippet`, however `kml:Snippet` may likely be supported by KML software implementations for an indeterminate period of time.

9.1.3.10.2 Content

Type:	xsd:string
Default Value:	none

9.1.3.11 kml:description

9.1.3.11.1 Description

Specifies a description of the *`kml:AbstractFeatureGroup`*. This should be displayed in the description balloon.

The text may include HTML content that is encoded as well-formed XML using HTML entity references or by enclosing the HTML within a CDATA section.

If the description includes the HTML `<a href="..." type="...">` tag, it should have an HTML `href` and `type` attribute and be interpreted as follows:

- The `href` attribute specifies a URL.

- If the target of the href is a KML resource, an earth browser should load the resource if the link is activated.

The `href` may reference another *`kml:AbstractFeatureGroup`* if its value is the fragment component of a URL (see 12.13.3.1). If such a link is activated the geographic view should fly to the *`kml:AbstractFeatureGroup`* whose ID matches the fragment. If this *`kml:AbstractFeatureGroup`* has a `kml:LookAt` or `kml:Camera` element, it shall be viewed from the specified viewpoint.

Special processing is required for embedded HTML links that append a display directive to the fragment component of the URL; such a directive (e.g. ";flyto") must be removed from the fragment component in order to obtain the shorthand pointer. Available directives are:

- ;flyto (default) – fly to the *`kml:AbstractFeatureGroup`*
- ;balloon – open the *`kml:AbstractFeatureGroup's`* balloon but do not fly to the *`kml:AbstractFeatureGroup`*
- ;balloonFlyto – open the *`kml:AbstractFeatureGroup's`* balloon and fly to the *`kml:AbstractFeatureGroup`*

For example, the following code indicates to open the resource *CraftsFairs.kml* resource, fly to the `kml:Placemark` whose ID is "Albuquerque," and open its balloon:

```
<description>
  <a href="http://myServer.com/CraftsFairs.kml#Albuquerque;balloonFlyto">
      One of the Best Art Shows in the West</a>
</description>
```

The `type` attribute specifies the MIME type for the target resource. An earth browser should interpret the target resource according to this specified MIME type when attempting to load it. To indicate that the target resource is KML specify the following MIME type:

```
type="application/vnd.google-earth.kml+xml"
```

To indicate that the target resource is a KMZ archive specify the following MIME type:

```
type="application/vnd.google-earth.kmz"
```

For example, the type attribute below indicates that an earth browser should attempt to load the target as a KML resource even though the file extension is *.php*:

```
<a href="myserver.com/cgi-bin/generate-kml.php#placemark123"
   type="application/vnd.google-earth.kml+xml"
```

9.1.3.11.2 Content

Type:	xsd:string
Default Value:	none

9.1.3.12 kml:AbstractViewGroup

An abstract placeholder for a viewpoint (`kml:Camera` or `kml:LookAt`).

See 14.1 AbstractViewGroup.

9.1.3.13 kml:AbstractTimePrimitiveGroup

An abstract placeholder for a time primitive (`kml:TimeStamp` or `kml:TimeSpan`).

See 15.1 kml:AbstractTimePrimitiveGroup.

9.1.3.14 kml:styleURL

9.1.3.14.1 Description

Specifies a reference to a `kml:Style` or `kml:StyleMap`. The reference shall be encoded as a URL with a fragment component (see 12.13.3.1). The value of the fragment shall be the **id** of a `kml:Style` or `kml:StyleMap` defined in a `kml:Document`.

See also 12.2 kml:Style; 12.3 kml:StyleMap, and 6.4 Shared Styles.

9.1.3.14.2 Content

Type:	xsd:anyURI
Default Value:	none

9.1.3.14.3 Example

To reference a `kml:Style` or `kml:StyleMap` in the same document:

```
<styleUrl>#myIconStyleID</styleUrl>
```

To reference a `kml:Style` or `kml:StyleMap` in a hosted document:

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

9.1.3.15 kml:AbstractStyleSelectorGroup

One or more `kml:Styles` or `kml:StyleMaps` used to style the *`kml:Feature`*.

See also 12.1 kml:AbstractStyleSelectorGroup and 6.4 Shared Styles.

9.1.3.16 kml:Region

Affects the visibility of the *`kml:AbstractFeatureGroup`* .

See 9.16 kml:Region.

9.1.3.17 kml:AbstractExtendedDataGroup

Allows for the addition of user-defined data.

See 9.2 kml:AbstractExtendedDataGroup.

9.1.3.18 kml:AbstractFeatureSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.1.3.19 kml:AbstractFeatureObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.1.4 Examples

9.1.4.1 Sample Use of HTML Elements within a Description

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
  <name>Feature.kml</name>
  <snippet>
    The snippet is a way ofproviding an alternative
    description that will beshown in the List view.
  </snippet>
  <description>
    <![CDATA[
      Styles: <i>Italics</i>, <b>Bold</b>, <u>Underlined</u>,
      <s>Strike Out</s>, subscript<sub>subscript</sub>,
      superscript<sup>superscript</sup>,
      <big>Big</big>, <small>Small</small>, <tt>Typewriter</tt>,
      <em>Emphasized</em>, <strong>Strong</strong>, <code>Code</code>
      <hr />
      Fonts:
      <font color="red">red by name</font>,
      <font color="#408010">leaf green by hexadecimal RGB</font>,
      <font size=1>size 1</font>, <font size=2>size 2</font>,
      <font size=3>size 3</font>, <font size=4>size 4</font>,
      <font size=5>size 5</font>, <font size=6>size 6</font>,
      <font size=7>size 7</font>,
      <font face=times>Times</font>,
      <font face=verdana>Verdana</font>,
      <font face=arial>Arial</font>
      <br/>
      <hr />
      Links:
      <a href="http://doc.trolltech.com/3.3/qstylesheet.html">
      QT Rich Text Rendering
      </a>
      <br />
      <hr />
      Alignment:
      <br />
      <p align=left>left</p><p align=center>center</p>
      <p align=right>right</p>
      <hr />
      Ordered Lists:
      <br />
      <ol><li>First</li><li>Second</li><li>Third</li></ol>
      <ol type="a"><li>First</li><li>Second</li><li>Third</li></ol>
      <ol type="A"><li>First</li><li>Second</li><li>Third</li></ol>
      <hr />
      Unordered Lists:
      <br />
      <ul><li>A</li><li>B</li><li>C</li></ul>
      <ul type="circle"><li>A</li><li>B</li><li>C</li></ul>
      <ul type="square"><li>A</li><li>B</li><li>C</li></ul>
      <hr />
      Definitions:
      <br />
      <dl>
      <dt>Scrumpy</dt>
      <dd>Hard English cider from the west country</dd>
      <dt>Pentanque</dt>
      <dd>A form of boules where the goal is to throw a metal ball as
      close as possible to a jack</dd>
      </dl>
      <hr />
      Block Quote:
      <br />
      <blockquote>
      We shall not cease from exploration<br />
      And the end of all our exploring<br />
      Will be to arrive where we started<br />
      And know the place for the first time
      </blockquote>
      <br />
      <hr />
      Centered:
      <br />
      <center>See, I have a Rhyme assisting<br />
      my feeble brain,<br />
      its tasks oft-times resisting!</center>
      <hr />
      Headings:
      <br />
      <h1>Header 1</h1>
      <h2>Header 2</h2>
      <h3>Header 3</h3>
      <h3>Header 4</h4>
      <h3>Header 5</h5>
      <hr />
      Images:
      <br />
      <img src="http://www.opengis.net/images/googleearth.gif" />
      <br />
      <i>Scaled image</i>
      <br />
      <img src="http://www.opengis.net/images/googleearth.gif"
      width="100" />
      <br />
      <hr />
      Tables:
      <table border="1" padding="3" width="300">
      <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
      <tr><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
      </table>
    ]]>
  </description>
  <Point>
    <coordinates>-122.378927,37.826793,0</coordinates>
  </Point>
</Placemark>
</kml>
```

9.1.4.2 Sample Use of Atom Elements

This example shows use of the `atom:author`, `atom:name` and `atom:link` attribution elements from the Atom namespace. In this case, `atom:author` and `atom:link` apply to both `kml:Placemarks`.

```
<kml xmlns="http://www.opengis.net/kml/2.2"
     xmlns:atom="http://www.w3.org/2005/Atom">
  <Document>
    <atom:author>
      <atom:name>J. K. Rowling</atom:name>
    </atom:author>
    <atom:link href="http://www.harrypotter.com" />
    <Placemark>
      <name>Hogwarts</name>
      <Point>
        <coordinates>1,1</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Little Hangleton</name>
      <Point>
        <coordinates>1,2</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

9.2 kml:AbstractExtendedDataGroup

9.2.1 Structure

```
<kml:AbstractExtendedDataGroup>...</kml:AbstractExtendedDataGroup>
```

9.2.2 Description

The following elements can be used wherever this abstract element is referenced:

- `kml:ExtendedData`
- `kml:Metadata` (deprecated)

`kml:AbstractExtendedDataGroup` is an abstract substitution head element that allows for the addition of user-defined data. Note that the `kml:Metadata` element was deprecated in KML 2.2, use `kml:ExtendedData` instead.

See also 9.3 kml:ExtendedData and 6.7.2.2 Complex Element Substitution.

9.2.3 Content

Type:	xsd:anyType
Default Value:	none

9.3 kml:ExtendedData

9.3.1 Structure

```
<kml:ExtendedData
 anyAttribute="anySimpleType [0..1]">
    <kml:Data>...</kml:Data> [0..*]
    <kml:SchemaData>...</kml:SchemaData> [0..*]
    <xsd:any/> (any element from foreign namespace with lax validation) [0..*]
</kml:ExtendedData>
```

9.3.2 Description

The `kml:ExtendedData` element offers three mechanisms for adding user-defined data to a *`kml:AbstractFeatureGroup`*. These mechanisms are:

- Adding arbitrary name/value data pairs using the `kml:Data` element (optionally typed using an `xsi:type` attribute)
- Adding instances of typed fields defined in the user-defined `kml:Schema` element
- Including any XML content defined in namespaces other than the KML namespace and null namespace

These mechanisms can be used concurrently within a single *`kml:AbstractFeatureGroup`* or KML document.

Child elements of `kml:ExtendedData` support entity substitution. See 6.5 Entity Replacement.

`kml:ExtendedData` should contain at least one child element outside of an update context, that is when not a descendant of `kml:Update`.

9.3.3 Content

9.3.3.1 kml:Data

An name/value pair with an optional typing mechanism (`xsi:type` attribute).

See 9.4 kml:Data.

9.3.3.2 kml:SchemaData

Encodes an instance of a user-defined data type defined by a referenced `kml:Schema`

See 9.5 kml:SchemaData.

9.3.3.3 xsd:any

9.3.3.3.1 Description

kml:ExtendedData may include any other well-formed, namespace-qualified XML content that is not from the KML namespace "http://www.opengis.net/kml/2.2"

9.3.3.3.2 Content

Type:	xsd:anyType
Default Value:	none

9.3.3.3.3 Example

The following example demonstrates the encoding of XML content from the "http://www.example.org/app" namespace:

```
<kml
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.opengis.net/kml/2.2"
  xmlns:app="http://www.example.org/app"
  xmlns:gml="http://www.opengis.net/gml">
  <Placemark>
    <name>A road</name>
    <ExtendedData>
      <app:Road>
        <app:numberOfLanes>2</app:numberOfLanes>
        <app:pavement>gravel</app:pavement>
      </app:Road>
    </ExtendedData>
  </Placemark>
</kml>
```

9.3.4 Attributes

9.3.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

9.4 kml:Data

9.4.1 Structure

```
<kml:Data
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]"
 name="string [0..1]"
 uom="anyURI [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:displayName>...</kml:displayName> [0..1]
    <kml:value>...</kml:value> [1]
    <kml:DataExtension>...</kml:DataExtension> [0..*]
</kml:Data>
```

9.4.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Creates a name/value data pair, where:

- The data pair is identified by the name attribute
- The value of the data pair is supplied by kml:value

Both name and kml:value should be encoded. The value of the name attribute shall be unique within the context of its parent kml:ExtendedData element. Since kml:value has the simple ur-type (xsd:anySimpleType), an xsi:type attribute can be provided on kml:value, to specify a simple datatype. If the xsi:type attribute is present, its value shall be a primitive XML Schema datatype; an XML Schema validator would then be able to enforce that kml:value is consistent with the datatype specified in the xsi:type attribute.

9.4.3 Content

9.4.3.1 kml:displayName

9.4.3.1.1 Description

An alternate display name.

9.4.3.1.2 Content

Type:	xsd:string
Default Value:	none

9.4.3.2 kml:value

9.4.3.2.1 Description

Value of the data pair.

9.4.3.2.2 Content

Type:	xsd: anySimpleType
Default Value:	none

9.4.3.3 kml:DataExtension

See 6.7.2.2 Complex Element Substitution.

9.4.4 Attributes

9.4.4.1 name

9.4.4.1.1 Description

Name of the data pair.

9.4.4.1.2 Content

Type:	xsd:string
Default Value:	none

9.4.4.2 uom

9.4.4.2.1 Description

The uom attribute indicates the unit of measure for some measured quantity. If the value is not an absolute URI (that refers to a definition, e.g. 'http://www.bipm.org/en/publications/si-brochure/metre.html'), it is expected to be a case-sensitive unit symbol (e.g. 'm' for meters, 's' for seconds, 'deg' for degrees, 'yd_us' for US yards) appearing in the Unified Code for Units of Measure (http://unitsofmeasure.org/ucum.html).

9.4.4.2.2 Content

Type:	xsd: anyURI
Default Value:	none

9.4.5 Example

In the following example, an `xsi:type` attribute is added to two of the `kml:value` elements and a `uom` attribute is added to one of the `kml:Data` elements.

```
<Placemark>
  <name>Club house</name>
  <ExtendedData>
    <Data name="holeNumber">
      <value>1</value>
    </Data>
    <Data name="holeYardage" uom="yd_us">
      <value xsi:type="xsd:double">234</value>
    </Data>
    <Data name="holePar">
      <value xsi:type="xsd:positiveInteger">4</value>
    </Data>
  </ExtendedData>
</Placemark>
```

9.5 kml:SchemaData

9.5.1 Structure

```
<kml:SchemaData
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]"
 schemaUrl="anyURI [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:SimpleData>...</kml:SimpleData> [0..*]
   <kml:SimpleArrayData>...</kml:SimpleArrayData> [0..*]
   <kml:SchemaDataExtension>...</kml:SchemaDataExtension> [0..*]
</kml:SchemaData>
```

9.5.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Encodes an instance of a user-defined data type defined by a referenced kml:Schema.

A kml:SchemaData element shall reference a kml:Schema element using the kml:schemaUrl attribute.

9.5.3 Content

9.5.3.1 kml:SimpleData

See 9.6 kml:SimpleData.

9.5.3.2 kml:SimpleArrayData

See 9.7 kml:SimpleArrayData.

9.5.3.3 kml:SchemaDataExtension

See 6.7.2.2 Complex Element Substitution.

9.5.4 Attributes

9.5.4.1 schemaUrl

9.5.4.1.1 Description

The value of `kml:schemaURL` should either be: a full URL, a reference to a `kml:Schema id` attribute defined in an external KML resource; or a reference to a `kml:Schema id` defined in the same KML resource.

9.5.4.1.2 Content

Type:	xsd:anyURI
Default Value:	none

9.5.4.1.3 Examples

All of the following URI values are acceptable:

```
schemaUrl="http://host.com/PlacesIHaveLived.kml#my-schema-id"
```

```
schemaUrl="AnotherFile.kml#my-schema-id"
```

```
schemaUrl="#schema-id"   <!-- same KML document -->
```

9.6 kml:SimpleData

9.6.1 Structure

```
<kml:SimpleData
 name="string [1]"
 anyAttribute="anySimpleType [0..1]">
    anySimpleType
</kml:SimpleData>
```

9.6.2 Description

Encodes an instance of a user-defined field defined by a referenced `kml:SimpleField`.

The value of `kml:SimpleData` shall be of the data type specified in the `type` attribute of the referenced `kml:SimpleField` ; the enforcement of this rule is possible, but not so simple using XML Schema validation. Methods for automated datatype consistency enforcement for `kml:SimpleData` values using both Schematron and XML Schema validation are discussed in([10], 8.2 Validate `kml:SimpleData` for Datatype Consistency),. `kml:SimpleData` is of complex type that extends from the simple ur-type `xsd:anySimpleType`, so an `xsi:type` attribute could optionally be provided to enable datatype enforcement using XML Schema validation. However, unlike the case of `kml:value`, the valid values of the `xsi:type` attribute would not be simple datatypes, rather they would be complex types that must be validly derived from `kml:SimpleDataType` (in a schema type definition outside the KML namespace.)

The required `name` attribute shall be used to identify the `kml:SimpleField` by name. The identified `kml:SimpleField` shall be declared within the `kml:Schema` element that is referenced from the `kml:SchemaURL` attribute.

9.6.3 Content

Type:	xsd: anySimpleType
Default Value:	none

9.6.4 Attributes

9.6.4.1 name

9.6.4.1.1 Description

A name value reference to the corresponding `kml:SimpleField`. See 9.11.4.2 name.

9.6.4.1.2 Content

Type: xsd:string
Default Value: none

9.6.4.2 anyAttribute

See 7.1.4.3 anyAttribute.

9.6.5 Example

User-defined data elements are shown in the following example:

```
<kml xmlns="http://earth.google.com/kml/2.2">
  <Document>
    <name>ExtendedData+SchemaData</name>
    <open>1</open>
    <!-- Create a balloon template referring to the user-defined type -->
    <Style id="trailhead-balloon-template">
      <BalloonStyle>
        <text>
          <![CDATA[
            <h2>My favorite trails!</h2>
            <br/><br/>
            The $[TrailHeadType/TrailHeadName/displayName] is
<i>$[TrailHeadType/TrailHeadName]</i>.
            The trail is $[TrailHeadType/TrailLength] miles.<br/>
            The climb is $[TrailHeadType/ElevationGain] meters.<br/><br/>
          ]]>
        </text>
      </BalloonStyle>
    </Style>

    <!-- Declare the type "TrailHeadType" with 3 fields -->
    <Schema name="TrailHeadType" id="TrailHeadTypeId">
      <SimpleField type="string" name="TrailHeadName">
        <displayName><![CDATA[<b>Trail Head Name</b>]]></displayName>
      </SimpleField>
      <SimpleField type="double" name="TrailLength">
        <displayName><![CDATA[<i>The length in miles</i>]]></displayName>
      </SimpleField>
      <SimpleField type="int" name="ElevationGain">
        <displayName><![CDATA[<i>change in altitude</i>]]></displayName>
      </SimpleField>
    </Schema>
    <!-- Instantiate some Placemarks extended with TrailHeadType fields -->
    <Placemark>
      <name>Easy trail</name>
      <styleUrl>#trailhead-balloon-template</styleUrl>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Pi in the sky</SimpleData>
          <SimpleData name="TrailLength">3.14159</SimpleData>
          <SimpleData name="ElevationGain">10</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-122.000,37.002</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Difficult trail</name>
      <styleUrl>#trailhead-balloon-template</styleUrl>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Mount Everest</SimpleData>
          <SimpleData name="TrailLength">347.45</SimpleData>
          <SimpleData name="ElevationGain">10000</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-121.998,37.0078</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

9.7 kml:SimpleArrayData

9.7.1 Structure

```
<kml:SimpleArrayData
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]"
 name="string [1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:value>...</kml:value> [0..*]
   <kml:SimpleArrayDataExtension>...</kml:SimpleArrayDataExtension> [0..*]
</kml:SimpleArrayData>
```

9.7.2 Description

Encodes an instance of a user-defined data array defined by a referenced kml:SimpleArrayField.

The kml:value children of kml:SimpleArrayData shall each be of the data type specified in the type attribute of the referenced kml:SimpleArrayField.

The required name attribute shall be used to identify the kml:SimpleArrayField by name. The identified kml:SimpleArrayField shall be declared within the kml:Schema element that is referenced from the kml:SchemaURL attribute.

9.7.3 Content

9.7.3.1 kml:value

See 9.4.3.2 kml:value.

9.7.3.2 kml:SimpleArrayDataExtension

See 6.7.2.2 Complex Element Substitution.

9.7.4 Attributes

9.7.4.1 name

9.7.4.1.1 Description

A name value reference to the corresponding `kml:SimpleArrayField`. See 9.12.4.2 name.

9.7.4.1.2 Content

Type: xsd:string
Default Value: none

9.7.4.2 anyAttribute

See 7.1.4.3 anyAttribute.

9.7.5 Example

See 10.15.4.2 Example 2 – Extended Data Arrays for an example in the context of `kml:Track`.

9.8 kml:AbstractContainerGroup

9.8.1 Structure

```
<kml:AbstractContainerGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:AbstractContainerSimpleExtensionGroup>...
     </kml:AbstractContainerSimpleExtensionGroup> [0..*]
    <kml:AbstractContainerObjectExtensionGroup>...
     </kml:AbstractContainerObjectExtensionGroup> [0..*]
</kml:AbstractContainerGroup>
```

9.8.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractFeatureGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:Document
- kml:Folder

For convenience in constructing KML feature hierarchies, unless overruled by the presence of such elements locally, the value of the following *kml:AbstractFeatureGroup* elements shall be inherited by all *kml:AbstractFeatureGroup* members of a feature hierarchy:

- atom:author
- atom:link
- kml:Region
- *kml:AbstractTimePrimitiveGroup*

Thus it is not necessary for a child *kml:AbstractFeatureGroup* to carry any of these elements where their local value is the same as that of its parent kml:Document or kml:Folder. Inheritance of these elements continues to any depth of nesting, but if overruled by a local declaration, then the new value is inherited by all its children in turn. Notwithstanding this rule, such elements may be used locally even if they have the same value as that of a parent kml:Document or kml:Folder.

9.8.3 Content

9.8.3.1 kml:AbstractContainerSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.8.3.2 kml:AbstractContainerObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.9 kml:Document

9.9.1 Structure

```
<kml:Document
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:AbstractContainerSimpleExtensionGroup>...
     </kml:AbstractContainerSimpleExtensionGroup> [0..*]
    <kml:AbstractContainerObjectExtensionGroup>...
     </kml:AbstractContainerObjectExtensionGroup> [0..*]
    <kml:Schema>...</kml:Schema> [0..*]
    <kml:AbstractFeatureGroup>...</kml:AbstractFeatureGroup> [0..*]
    <kml:DocumentSimpleExtensionGroup>...</kml:DocumentSimpleExtensionGroup>
[0..*]
    <kml:DocumentObjectExtensionGroup>...</kml:DocumentObjectExtensionGroup>
[0..*]
</kml:Document>
```

9.9.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractContainerGroup*

A `kml:Document` is a container for KML features, shared styles, and user-defined schemas.

See also 6.4 Shared Styles.

9.9.3 Content

9.9.3.1 kml:Schema

Specifies a user-defined schema.

See 9.10 kml:Schema.

9.9.3.2 kml:AbstractFeatureGroup

See 9.1 kml:AbstractFeatureGroup.

9.9.3.3 kml:DocumentSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.9.3.4 kml:DocumentObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.10 kml:Schema

9.10.1 Structure

```
<kml:Schema
 name="string [0..1]"
 id="ID [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:SimpleField>...</kml:SimpleField> [0..*]
    <kml:SimpleArrayField>...</kml:SimpleArrayField> [0..*]
    <kml:SchemaExtension>...</kml:SchemaExtension> [0..*]
</kml:Schema>
```

9.10.2 Description

Specifies a user-defined schema that is used to add user-defined data encoded within a child `kml:ExtendedData` element of a *`kml:AbstractFeatureGroup`*. `kml:Schema` shall have an `id` so that instances of it (encoded as `kml:SchemaData`) may reference it.

9.10.3 Content

9.10.3.1 kml:SimpleField

See 9.11 kml:SimpleField.

9.10.3.2 kml:SimpleArrayField

See 9.12 kml:SimpleArrayField.

9.10.3.3 kml:SchemaExtension

See 6.7.2.2 Complex Element Substitution.

9.10.4 Attributes

9.10.4.1 id

9.10.4.1.1 Description

Unique identifier of the `kml:AbstractObjectGroup` within the KML document instance.

9.10.4.1.2 Content

Type:	xsd:ID
Default Value:	none

9.10.4.2 name

See 9.6.4.1 name.

9.10.4.3 anyAttribute

See 7.1.4.3 anyAttribute.

9.10.5 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <Schema name="TrailHeadType" id="TrailHeadTypeId">
    <SimpleField type="string" name="TrailHeadName">
      <displayName><![CDATA[<b>Trail Head Name</b>]]></displayName>
    </SimpleField>
    <SimpleField type="double" name="TrailLength">
      <displayName><![CDATA[<i>The length in miles</i>]]></displayName>
    </SimpleField>     <SimpleField type="int" name="ElevationGain">
      <displayName><![CDATA[<i>change in altitude</i>]]></displayName>
    </SimpleField>
  </Schema>
</Document>
</kml>
```

9.11 kml:SimpleField

9.11.1 Structure

```
<kml:SimpleField
 type="string [0..1]"
 name="string [0..1]"
 uom="anyURI [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:displayName>...</kml:displayName> [0..1]
    <kml:SimpleFieldExtension>...</kml:SimpleFieldExtension> [0..*]
</kml:SimpleField>
```

9.11.2 Description

Specifies a user-defined data field. The `name` and `type` attributes shall be specified. The main purpose of `kml:SimpleField` is to enable KML implementations to efficiently store a data value in memory as a specified data type (e.g. storing an integer value as an integer datatype rather than a string representation of the integer). The datatype information encoded in the `type` attribute of `kml:SimpleField` can also be used to enable automated datatype consistency checking between the `type` attribute and a corresponding `kml:SimpleData` value. For example, a Schematron assertion can provide such a datatype consistency check as discussed in ([10], 8.2 Validate `kml:SimpleData` for Datatype Consistency).

9.11.3 Content

9.11.3.1 kml:displayName

See 12.8 kml:displayName.

9.11.3.2 kml:SimpleFieldExtension

See 6.7.2.2 Complex Element Substitution.

9.11.4 Attributes

9.11.4.1 type

9.11.4.1.1 Description

Type of the field. The `type` content shall specify a qualified name of an XML Schema simple type. Simple types include the XML Schema primitive datatypes and any type that is validly derived from a simple XML Schema datatype (e.g. `kml:angle90Type`). The following XML Schema datatypes are also valid example values of `type`:

- xsd:string
- xsd:unsignedInt
- xsd:unsignedShort
- xsd:double
- xsd:int
- xsd: short
- xsd: float
- xsd: boolean
- xsd: date
- xsd: dateTime

9.11.4.1.2 Content

Type: xsd:string
Default Value: none

9.11.4.2 name

Name that identifies the `kml:SimpleField`. Used as a reference value in the `kml:SimpleData` `name` attribute. See 9.6.4.1 name.

9.11.4.3 uom

Unit of measure symbol for a data value representing a measured quantity. See 9.4.4.2 uom.

9.11.4.4 anyAttribute

See 7.1.4.3 anyAttribute.

9.12 kml:SimpleArrayField

9.12.1 Structure

```
<kml:SimpleArrayField
 type="string [0..1]"
 name="string [0..1]"
 uom="anyURI [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:displayName>...</kml:displayName> [0..1]
    <kml:SimpleArrayFieldExtension>...</kml:SimpleArrayFieldExtension> [0..*]
</kml:SimpleArrayField>
```

9.12.2 Description

Specifies a user-defined data field for an array of data (similar to 9.11 kml:SimpleField in both structure and purpose). The `name` and `type` attributes shall be specified.

9.12.3 Content

9.12.3.1 kml:displayName

See 12.8 kml:displayName.

9.12.3.2 kml:SimpleArrayFieldExtension

See 6.7.2.2 Complex Element Substitution.

9.12.4 Attributes

9.12.4.1 type

Type of the field. See 9.11.4.1 type.

9.12.4.2 name

Name that identifies the `kml:SimpleArrayField`. Used as a reference value in the `kml:SimpleArrayData` `name` attribute.

9.12.4.3 uom

Unit of measure symbol for a data value representing a measured quantity.

See 9.4.4.2 uom.

9.12.4.4 anyAttribute

See 7.1.4.3 anyAttribute.

9.13 kml:Folder

9.13.1 Structure

```
<kml:Folder
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:AbstractContainerSimpleExtensionGroup>...
     </kml:AbstractContainerSimpleExtensionGroup> [0..*]
    <kml:AbstractContainerObjectExtensionGroup>...
     </kml:AbstractContainerObjectExtensionGroup> [0..*]
    <kml:AbstractFeatureGroup>...</kml:AbstractFeatureGroup> [0..*]
    <kml:FolderSimpleExtensionGroup>...</kml:FolderSimpleExtensionGroup> [0..*]
    <kml:FolderObjectExtensionGroup>...</kml:FolderObjectExtensionGroup> [0..*]
</kml:Folder>
```

9.13.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractContainerGroup*

A kml:Folder is used to organize *kml:AbstractFeatureGroup* elements hierarchically. Contains zero or more *kml:AbstractFeatureGroup* elements and allows the creation of one or more nested hierarchies of KML features.

kml:Folder should contain at least one child element outside of an update context, that is when not a descendant of kml:Update.

9.13.3 Content

9.13.3.1 kml:AbstractFeatureGroup

See 9.1 kml:AbstractFeatureGroup.

9.13.3.2 kml:FolderSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.13.3.3 kml:FolderObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.13.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Folder>
  <name>Folder.kml</name>
  <open>1</open>
  <description>
    A folder is a container that can hold multiple other objects
  </description>
  <Placemark>
    <name>Folder object 1 (Placemark)</name>
    <Point>
      <coordinates>-122.377588,37.830266,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>Folder object 2 (Polygon)</name>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.377830,37.830445,0
            -122.377576,37.830631,0
            -122.377840,37.830642,0
            -122.377830,37.830445,0
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark>
  <Placemark>
    <name>Folder object 3 (Path)</name>
    <LineString>
      <tessellate>1</tessellate>
      <coordinates>
        -122.378009,37.830128,0 -122.377885,37.830379,0
      </coordinates>
    </LineString>
  </Placemark>
</Folder>
</kml>
```

9.14 kml:Placemark

9.14.1 Structure

```
<kml:Placemark
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:AbstractGeometryGroup>...</kml:AbstractGeometryGroup> [0..1]
    <kml:PlacemarkSimpleExtensionGroup>...
     </kml:PlacemarkSimpleExtensionGroup> [0..*]
    <kml:PlacemarkObjectExtensionGroup>...
     </kml:PlacemarkObjectExtensionGroup> [0..*]
</kml:Placemark>
```

9.14.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractFeatureGroup*

A kml:Placemark is a *kml:AbstractFeatureGroup* with an associated *kml:AbstractGeometryGroup*.

A kml:Placemark with a kml:Point geometry should be drawn with an icon to mark the kml:Placemark in the geographic view. The point itself determines the position of the kml:Placemark's name and display icon.

kml:Placemark should contain a kml:*AbstractGeometryGroup* element outside of an update context, that is when not a descendant of kml:Update.

9.14.3 Content

9.14.3.1 kml:AbstractGeometryGroup

See 10.1 kml:AbstractGeometryGroup.

9.14.3.2 kml:PlacemarkSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.14.3.3 kml:PlacemarkObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.14.4 Example

```
<Placemark>
  <name>New Placemark</name>
  <description>Some Descriptive text.</description>
  <LookAt>
    <longitude>-90.86879847669974</longitude>
    <latitude>48.25330383601299</latitude>
    <range>440.8</range>
    <tilt>8.3</tilt>
    <heading>2.7</heading>
  </LookAt>
  <Point>
    <coordinates>-90.86948943473118,48.25450093195546,0</coordinates>
  </Point>
</Placemark>
```

9.15 kml:NetworkLink

9.15.1 Structure

```
<kml:NetworkLink
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:refreshVisibility>...</kml:refreshVisibility> [0..1]
    <kml:flyToView>...</kml:flyToView> [0..1]
    <kml:Link>...</kml:Link> [0..1]
    <kml:NetworkLinkSimpleExtensionGroup>...
     </kml:NetworkLinkSimpleExtensionGroup> [0..*]
    <kml:NetworkLinkObjectExtensionGroup>...
     </kml:NetworkLinkObjectExtensionGroup> [0..*]
</kml:NetworkLink>
```

9.15.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractFeatureGroup*

References a KML resource on a local or remote network. kml:NetworkLink elements may be used in combination with kml:Region elements to efficiently load and display large datasets.

9.15.3 Content

9.15.3.1 kml:refreshVisibility

9.15.3.1.1 Description

Specifies the control over the visibility of any *kml:AbstractFeatureGroup* elements within the referenced KML resource.

A value of 0 or false shall leave the visibility of any referenced *kml:AbstractFeatureGroup* elements in the geographic view within the control of the earth browser user.

A value of 1 or true shall require any referenced *kml:AbstractFeatureGroup* elements to be visible within the geographic view whenever such *kml:AbstractFeatureGroup* elements are refreshed.

9.15.3.1.2 Content

Type:	xsd:boolean
Default Value:	0 or false

9.15.3.2 kml:flyToView

9.15.3.2.1 Description

Specifies whether to adjust the geographic view upon kml:NetworkLink activation.

A value of 0 or false indicates that the geographic view shall remain unchanged. A value of 1 or true indicates that the geographic view shall be displayed according to the *kml:AbstractViewGroup* specified by either:

- a kml:NetworkLinkControl
- a child *kml:AbstractFeatureGroup* of kml:kml

if they exist in the referenced KML resource. The *kml:AbstractViewGroup* of the kml:NetworkLinkControl shall take precedence over the *kml:AbstractViewGroup* of the *kml:AbstractFeatureGroup* if they both exist. If neither exists then the view shall remain unchanged.

9.15.3.2.2 Content

Type:	xsd:boolean

Default Value: 0 or false

9.15.3.3 kml:Link

Specifies the location of the KML resource fetched by `kml:NetworkLink`. Note that `kml:Link` replaces the `kml:Url` element, which was deprecated in KML 2.2.

See 13.1 kml:Link, kml:Icon (kml:LinkType).

9.15.3.4 kml:NetworkLinkSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.15.3.5 kml:NetworkLinkObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.15.4 Example

```
<Document>
  <visibility>1</visibility>
  <NetworkLink>
    <name>NE US Radar</name>
    <refreshVisibility>1</refreshVisibility>
    <flyToView>1</flyToView>
    <Link>...</Link>
  </NetworkLink>
</Document>
```

9.16 kml:Region

9.16.1 Structure

```
<kml:Region
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentGroup>...</kml:AbstractExtentGroup> [0..1]
    <kml:Lod>...</kml:Lod> [0..1]
    <kml:RegionSimpleExtensionGroup>...</kml:RegionSimpleExtensionGroup> [0..*]
    <kml:RegionObjectExtensionGroup>...</kml:RegionObjectExtensionGroup> [0..*]
</kml:Region>
```

9.16.2 Description

This element can be used wherever the following element is referenced:

- *`kml:AbstractObjectGroup`*

Affects the visibility of the *`kml:AbstractFeatureGroup`*. A *`kml:AbstractFeatureGroup`* associated with a `kml:Region` is drawn only when the `kml:Region` is active.

A kml:Region affects visibility of a *kml:AbstractFeatureGroup*. kml:Regions define both culling and level-of-detail control over the display of the *kml:AbstractFeatureGroup*. A region shall specify a kml:LatLonAltBox element that describes an area of interest defined by geographic coordinates and altitudes. In addition, a kml:Region contains a kml:Lod element that defines a validity range of the associated kml:Region in terms of projected screen size.

Regions are inherited through a *kml:AbstractFeatureGroup* hierarchy and affect the visibility of *kml:AbstractFeatureGroup* elements that are defined lower in the hierarchy.

A kml:Region is said to be "active" when the bounding box is within the user's view and the LOD requirements are met. *kml:AbstractFeatureGroup* elements associated with a kml:Region are drawn only when the kml:Region is active. When the kml:viewRefreshMode is **onRegion**, the kml:Link or kml:Icon is loaded only when the kml:Region is active. In a *kml:AbstractContainerGroup* or kml:NetworkLink hierarchy, this calculation uses the kml:Region that is the closest ancestor in the hierarchy. See also 9.8 kml:AbstractContainerGroup regarding the inheritance of kml:Region within KML feature hierarchies.

kml:Region shall contain the kml:LatLonAltBox and kml:Lod child elements outside of an update context, that is when not a descendant of kml:Update.

9.16.3 Content

9.16.3.1 kml:AbstractExtentGroup

An abstract placeholder for kml:LatLonAltBox in the context of kml:Region.

See 9.17 kml:AbstractExtentGroup and 9.19 kml:LatLonAltBox.

9.16.3.2 kml:Lod

See 9.22 kml:Lod.

9.16.3.3 kml:RegionSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.16.3.4 kml:RegionObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.16.4 Example

```
<Region>
  <LatLonAltBox>
    <north>50.625</north>
    <south>45</south>
    <east>28.125</east>
    <west>22.5</west>
    <minAltitude>10</minAltitude>
    <maxAltitude>50</maxAltitude>
  </LatLonAltBox>
  <Lod>
    <minLodPixels>128</minLodPixels>
    <maxLodPixels>1024</maxLodPixels>
    <minFadeExtent>128</minFadeExtent>
    <maxFadeExtent>128</maxFadeExtent>
  </Lod>
</Region>
```

9.17 kml:AbstractExtentGroup

9.17.1 Structure

```
<kml:AbstractExtentGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentSimpleExtensionGroup>...
     </kml:AbstractExtentSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentObjectExtensionGroup>...
     </kml:AbstractExtentObjectExtensionGroup> [0..*]
</kml:AbstractExtentGroup>
```

9.17.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- *kml:AbstractLatLonBox*
- kml:LatLonQuad

9.17.3 Content

9.17.3.1 kml:AbstractExtentSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.17.3.2 kml:AbstractExtentObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.18 kml:AbstractLatLonBoxGroup

9.18.1 Structure

```
<kml:AbstractLatLonBoxGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentSimpleExtensionGroup>...
     </kml:AbstractExtentSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentObjectExtensionGroup>...
     </kml:AbstractExtentObjectExtensionGroup> [0..*]
    <kml:north>...</kml:north> [0..1]
    <kml:south>...</kml:south> [0..1]
    <kml:east>...</kml:east> [0..1]
    <kml:west>...</kml:west> [0..1]
    <kml:AbstractLatLonBoxSimpleExtensionGroup>...
     </kml:AbstractLatLonBoxSimpleExtensionGroup> [0..*]
    <kml:AbstractLatLonBoxObjectExtensionGroup>...
     </kml:AbstractLatLonBoxObjectExtensionGroup> [0..*]
</kml:AbstractLatLonBoxGroup>
```

9.18.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractExtentGroup*

The following elements can be used wherever this abstract element is referenced:

- *kml:LatLonAltBox*
- *kml:LatLonBox*

The `kml:AbstractLatLonBoxGroup` element represents a bounding box, which is spanned by latitude and longitude values satisfying: `kml:south` ≤ latitude ≤ `kml:north` and `kml:west` ≤ longitude ≤ `kml:east`.

9.18.3 Content

9.18.3.1 kml:north

9.18.3.1.1 Description

Specifies the latitude of the north edge of the bounding box, in decimal degrees in the interval: -180 < latitude ≤ 180. Note that the value of `kml:north` ≠ -180 because it must be strictly greater than the value of `kml:south`. Note also that values of |latitude| > 90 are atypical, but allowed for backwards compatibility with KML 2.2. The default value of `kml:north` is 90 (changed from the default value of 180 in KML 2.2). See Section 9.14.3.7 for additional information on value constraints.

9.18.3.1.2 Content

Type:	kml:angle180Type
Default Value:	90.0

9.18.3.2 kml:south

9.18.3.2.1 Description

Specifies the latitude of the south edge of the bounding box, in decimal degrees in the interval: -180 ≤ latitude < 180. Note that the value of `kml:south` ≠ 180 because it must be strictly less than the value of `kml:north`. Note also that values of |latitude| > 90 are atypical, but allowed for backwards compatibility with KML 2.2. The default value of `kml:south` is -90 (changed from the default value of -180 in KML 2.2).See Section 9.14.3.7 for additional information on value constraints.

9.18.3.2.2 Content

Type:	kml:angle180Type
Default Value:	-90.0

9.18.3.3 kml:east

9.18.3.3.1 Description

Specifies the longitude of the east edge of the bounding box, in decimal degrees in the interval: -360 < longitude ≤ 360. Note that the value of `kml:east` ≠ -360 because it must be strictly greater than the value of `kml:west`. See Section 9.14.3.7 for additional information on value constraints.

9.18.3.3.2 Content

Type:	kml:angle360Type
Default Value:	180.0

9.18.3.4 kml:west

9.18.3.4.1 Description

Specifies the longitude of the west edge of the bounding box, in decimal degrees in the interval: -360 < longitude < 360. Note that the value of `kml:west` ≠ 360 because it must be strictly less than the value of `kml:east` and also `kml:west` ≠ -360, because of the *uniqueness* and *non-self-overlap* constraints. See Section 9.14.3.7 for additional information on value constraints.

9.18.3.4.2 Content

Type:	kml:angle360Type
Default Value:	-180.0

9.18.3.5 kml:AbstractLatLonBoxSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.18.3.6 kml:AbstractLatLonBoxObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.18.3.7 Constraints

In KML 2.3, the allowed value range in decimal degrees used by `kml:east` and `kml:west` is extended by a factor of 2 (from ±180 in KML 2.2) to ±360. This was done in order to accommodate bounding boxes anywhere on the earth, including overlaps of the anti-meridian, and of any size up to full global coverage. With the extension of the longitude range, all degree values, except -360 = 0 = 360 (mod 360), have exactly two equivalent choices modulo 360, e.g. -359 = 1 (mod 360). The latitude range for kml:north and kml:south remain the same as in KML 2.2 and the following constraints `C1` (i.e. the *non-trivial latitude interval* constraint) and `C2` (i.e. the *non-trivial longitude interval* constraint) are unchanged:

`C1 kml:south < kml:north` (non-trivial latitude interval)
`C2 kml:west < kml:east` (non-trivial longitude interval)

New constraints in KML 2.3 are introduced with the longitude range extension to avoid self overlaps and to preserve uniqueness of longitude interval values:

`C3 kml:east` - `kml:west` ≤ 360 (non-self-overlap)

`C4` If (|`kml:west`| or |`kml:east`|) > 180, then `kml:east` > 0 and `kml:west` < 180 (uniqueness)

The constraint `C3` ensures that the longitude interval does not overlap itself. The constraint `C4` ensures the choice of the `kml:west` and `kml:east` values are unique for every longitude interval. Examples of allowed and prohibited pairs of `kml:west` and `kml:east` values are shown in Figure 9.

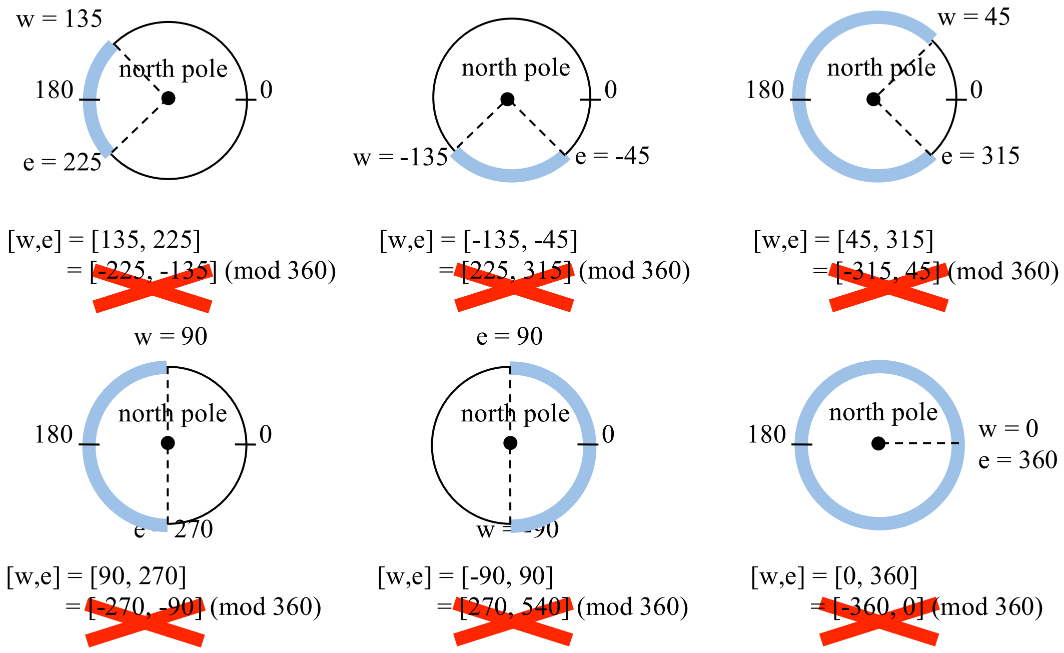


Figure 9: Uniqueness of longitude intervals from west to east ([w, e])

Additional examples of valid longitude intervals with full global coverage, starting and ending at different longitude values are shown in Figure 10.

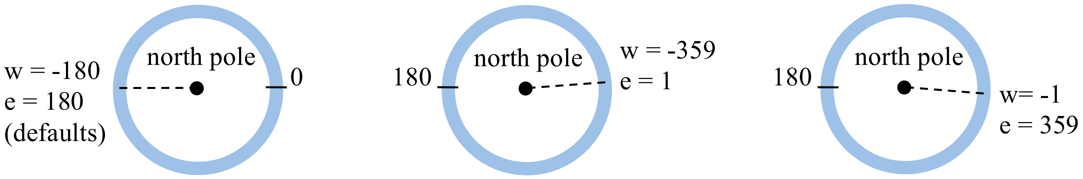


Figure 10: Longitude intervals with full global coverage and various begin/end values

9.19 kml:LatLonAltBox

9.19.1 Structure

```
<kml:LatLonAltBox
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentSimpleExtensionGroup>...
     </kml:AbstractExtentSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentObjectExtensionGroup>...
     </kml:AbstractExtentObjectExtensionGroup> [0..*]
    <kml:north>...</kml:north> [0..1]
    <kml:south>...</kml:south> [0..1]
    <kml:east>...</kml:east> [0..1]
    <kml:west>...</kml:west> [0..1]
    <kml:AbstractLatLonBoxSimpleExtensionGroup>...
     </kml:AbstractLatLonBoxSimpleExtensionGroup> [0..*]
    <kml:AbstractLatLonBoxObjectExtensionGroup>...
     </kml:AbstractLatLonBoxObjectExtensionGroup> [0..*]
    <kml:minAltitude>...</kml:minAltitude> [0..1]
    <kml:maxAltitude>...</kml:maxAltitude> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:LatLonAltBoxSimpleExtensionGroup>...
     </kml:LatLonAltBoxSimpleExtensionGroup> [0..*]
    <kml:LatLonAltBoxObjectExtensionGroup>...
     </kml:LatLonAltBoxObjectExtensionGroup> [0..*]
</kml:LatLonAltBox>
```

9.19.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

NOTE: The schema type for `kml:LatLonAltBox` derives by extension from the schema type for `kml:AbstractLatLonAltBoxGroup`, even though it does not substitute for it.

Specifies the extent of a 3D bounding box, such that:

- `kml:north` is greater than `kml:south`;
- `kml:east` is greater than `kml:west`;
- `kml:minAltitude` is less than or equal to `kml:maxAltitude`;
- if `kml:minAltitude` and `kml:maxAltitude` are both present, `kml:altitudeMode` shall not have a value of **clampToGround**.

9.19.3 Content

9.19.3.1 minAltitude

9.19.3.1.1 Description

Specified in meters above the vertical datum (and is affected by the `kml:altitudeMode` specification).

9.19.3.1.2 Content

Type:	xsd: double
Default Value:	0.0

9.19.3.2 maxAltitude

9.19.3.2.1 Description

Specified in meters above the vertical datum (and is affected by the `kml:altitudeMode` specification).

9.19.3.2.2 Content

Type:	xsd: double
Default Value:	0.0

9.19.3.3 kml:altitudeMode

See 9.20 kml:altitudeMode.

9.19.3.4 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

9.19.3.5 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.19.3.6 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.19.3.7 kml:LatLonAltBoxSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.19.3.8 kml:LatLonAltBoxObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.19.4 Example

```
<LatLonAltBox>
  <north>43.374</north>
  <south>42.983</south>
  <east>-0.335</east>
  <west>-1.423</west>
  <minAltitude>0</minAltitude>
  <maxAltitude>0</maxAltitude>
</LatLonAltBox>
```

9.20 kml:altitudeMode

9.20.1 Structure

```
<kml:altitudeMode>...</kml:altitudeMode>
```

9.20.2 Description

Specifies how the altitude components are interpreted in `kml:coordinates`, `kml:coord` and `kml:altitude` elements. The allowed values of `kml:altitudeMode` are:

- **clampToGround** - (default) The altitude value is ignored and the position is projected onto the ground.
- **relativeToGround** - Sets the altitude of the element relative to the actual ground elevation of a particular location. For example, 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 elevation for the icon of a point placemark elevation is 9 meters with this mode. However, if the same coordinate is set over a location where the ground elevation is 10 meters above sea level, then the elevation of the coordinate is 19 meters above sea level.
- **absolute** - Sets the altitude of the coordinate relative to sea level, regardless of the actual elevation of the terrain beneath the element. For example, if you set the altitude of a coordinate to 10 meters with an **absolute** altitude mode, the icon of a point placemark will appear to be at ground level if the terrain beneath is also 10 meters above sea level. If the terrain is 3 meters above sea level, the placemark will appear elevated above the terrain by 7 meters.

If `kml:altitudeMode` is set to **relativeToGround** or **absolute** then the associated altitude component(s) should be present within the element to which it applies.

See also 6.3 Geometry Interpolation for 3D Earth Browsers.

9.20.3 Content

Type:	kml:altitudeModeEnumType
Default Value:	**clampToGround**

9.21 kml:seaFloorAltitudeMode

9.21.1 Structure

```
<kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode>
```

9.21.2 Description

Specifies how the altitude components are interpreted in `kml:coordinates`, `kml:coord` and `kml:altitude` elements. The allowed values of `kml:seaFloorAltitudeMode` are:

- **relativeToSeaFloor** - Interprets the altitude as a value in meters above the sea floor. If the point is above land rather than sea, the altitude will be interpreted as being above the ground.
- **clampToSeaFloor** – The specified altitude value is ignored, and the point will be projected onto the sea floor. If the point is on land rather than at sea, the point will be positioned on the ground.

9.21.3 Content

Type:	kml:seaFloorAltitudeModeEnumType
Default Value:	none

9.22 kml:Lod

9.22.1 Structure

```
<kml:Lod
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:minLodPixels>...</kml:minLodPixels> [0..1]
    <kml:maxLodPixels>...</kml:maxLodPixels> [0..1]
    <kml:minFadeExtent>...</kml:minFadeExtent> [0..1]
    <kml:maxFadeExtent>...</kml:maxFadeExtent> [0..1]
    <kml:LodSimpleExtensionGroup>...</kml:LodSimpleExtensionGroup> [0..*]
    <kml:LodObjectExtensionGroup>...</kml:LodObjectExtensionGroup> [0..*]
</kml:Lod>
```

9.22.2 Description

This element can be used wherever the following element is referenced:

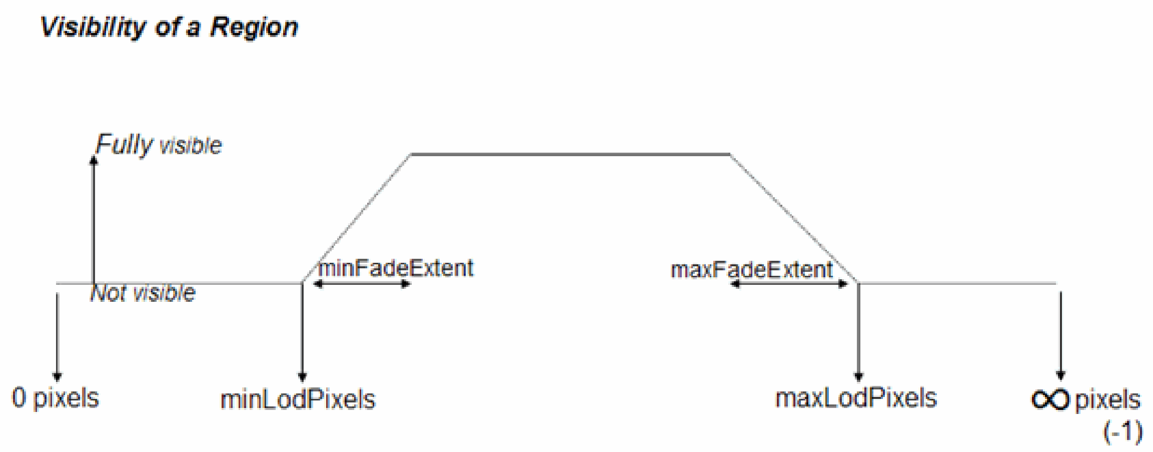
- *kml:AbstractObjectGroup*

Specifies the level of detail to use when displaying a `kml:Region`.

An earth browser should calculate the size of the `kml:Region` when projected onto screen space then compute the square root of the `kml:Region's` area. For example, if an untiled `kml:Region` is square and the viewpoint is directly above the `kml:Region`, this measurement is equal to the width of the projected `kml:Region`. If this measurement falls within the limits defined by kml:minLodPixels and kml:maxLodPixels, and if the kml:LatLonAltBox is in view, then the kml:Region should be activated. If this limit is not reached, the associated geometry should not be drawn since it would be too far from the user's viewpoint to be visible.

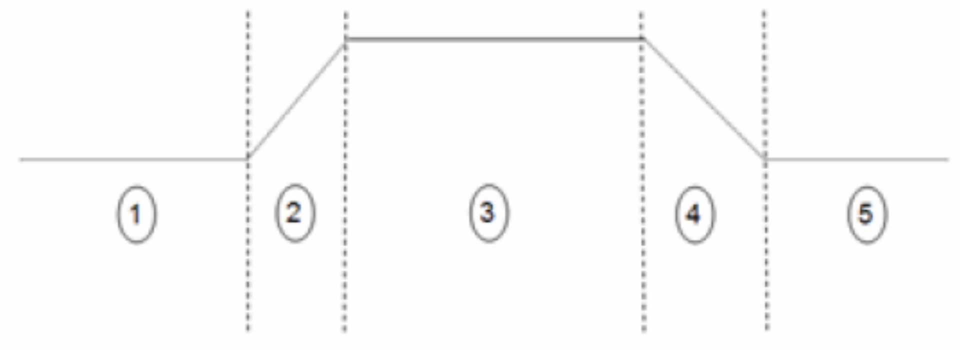
kml:minLodPixels shall be less than kml:maxLodPixels (where a value of -1 = infinite). It is also advised that kml:minFadeExtent + kml:maxFadeExtent is less than or equal to kml:maxLodPixels - kml:minLodPixels.

The following diagram demonstrates how kml:Lod is used in the determination of the visibility of a region:



In the following diagram, if *P*=the calculated projected pixel size, the circled numbers indicate the following:

```
if  (P < minLodPixels)
  opacity=0                                 //#1 in diagram
else if(P < minLodPixels + minFadeExtent)
  opacity=(P - minLodPixels)/minFadeExtent  //#2 in diagram
else if  (P < maxLodPixels - maxFadeExtent)
  opacity=1                                 //#3 in diagram
else if  (P < maxLodPixels)
  opacity=(maxLodPixels-P)/maxFadeExtent    //#4 in diagram
else
  opacity=0                                 //#5 in diagram
```



`kml:Lod` shall contain the `kml:minLodPixels` outside of an update context, that is when not a descendant of `kml:Update`.

See also 9.16 kml:Region.

9.22.3 Content

9.22.3.1 kml:minLodPixels

9.22.3.1.1 Description

Measurement in screen pixels that represents the minimum limit of the visibility range for a given `kml:Region`.

9.22.3.1.2 Content

Type:	xsd:double
Default Value:	0.0

9.22.3.2 kml:maxLodPixels

9.22.3.2.1 Description

Measurement in screen pixels that represents the maximum limit of the visibility range for a given `kml:Region`. The default value of −1.0 indicates "active to infinite size."

9.22.3.2.2 Content

Type:	xsd:double
Default Value:	-1.0

9.22.3.3 kml:minFadeExtent

9.22.3.3.1 Description

Distance over which the geometry fades, from fully opaque to fully transparent. This ramp value, expressed in screen pixels, is applied at the minimum end of the LOD (visibility) limits.

9.22.3.3.2 Content

Type:	xsd:double
Default Value:	0.0

9.22.3.4 kml:maxFadeExtent

9.22.3.4.1 Description

Distance over which the geometry fades, from fully transparent to fully opaque. This ramp value, expressed in screen pixels, is applied at the maximum end of the LOD (visibility) limits.

9.22.3.4.2 Content

Type:	xsd:double
Default Value:	0.0

9.22.3.5 kml:LodSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.22.3.6 kml:LodObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.22.4 Example

```
<Region>
  <LatLonAltBox>
    <north>50.625</north>
    <south>45</south>
    <east>28.125</east>
    <west>22.5</west>
    <minAltitude>10</minAltitude>
    <maxAltitude>50</maxAltitude>
  </LatLonAltBox>
  <Lod>
    <minLodPixels>128</minLodPixels>
    <maxLodPixels>1024</maxLodPixels>
    <minFadeExtent>128</minFadeExtent>
    <maxFadeExtent>128</maxFadeExtent>
  </Lod>
</Region>
```

9.23 kml:Tour

9.23.1 Structure

```
<kml:Tour
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:name>...</kml:name> [0..1]
   <kml:visibility>...</kml:visibility> [0..1]
   <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
   <kml:open>...</kml:open> [0..1]
   <atom:author>...</atom:author> [0..1]
   <atom:link>...</atom:link> [0..1]
   <kml:address>...</kml:address> [0..1]
```

```
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:AbstractSnippetGroup>...</kml:AbstractSnippetGroup> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml:AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:Playlist>...</kml:Playlist> [0..1]
    <kml:TourSimpleExtensionGroup>...</kml:TourSimpleExtensionGroup> [0..*]
    <kml:TourObjectExtensionGroup>...</kml:TourObjectExtensionGroup> [0..*]
</kml:Tour>
```

9.23.2 Description

A kml:Tour specifies a controlled flight through a series of geographic locations. The aspects of the flight that can be controlled include: the duration between locations, the mode of flight (smooth or bounce), what and when sound tracks should be played, and how KML features are updated throughout the tour. The kml:Tour is comprised of a playlist, which contains tour primitives, such as fly-to viewpoints, flight durations, pauses, sound cues, and KML updates. The tour primitives can be played in series or in parallel and are used to specify the behavior of the KML browser along different segments of the tour timeline.

9.23.3 Content

9.23.3.1 kml:Playlist

Contains any number of tour primitives.

See 9.24 kml:Playlist.

9.23.3.2 kml:TourSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.23.3.3 kml:TourObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.23.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>A tour and some features</name>
    <open>1</open>
    <Tour>
      <name>Play me!</name>
      <Playlist>
```

```
        <FlyTo>
          <duration>5.0</duration>
          <!-- bounce is the default flyToMode -->
          <Camera>
            <longitude>170.157</longitude>
            <latitude>-43.671</latitude>
            <altitude>9700</altitude>
            <heading>-6.333</heading>
            <tilt>33.5</tilt>
          </Camera>
        </FlyTo>
        <Wait>
          <duration>1.0</duration>
        </Wait>
        <FlyTo>
          <duration>6.0</duration>
          <Camera>
            <longitude>174.063</longitude>
            <latitude>-39.663</latitude>
            <altitude>18275</altitude>
            <heading>-4.921</heading>
            <tilt>65</tilt>
            <altitudeMode>absolute</altitudeMode>
          </Camera>
        </FlyTo>
        <FlyTo>
          <duration>3.0</duration>
          <flyToMode>smooth</flyToMode>
          <LookAt>
            <longitude>174.007</longitude>
            <latitude>-39.279</latitude>
            <altitude>0</altitude>
            <heading>112.817</heading>
            <tilt>68.065</tilt>
            <range>6811.884</range>
            <altitudeMode>relativeToGround</altitudeMode>
          </LookAt>
        </FlyTo>
        <FlyTo>
          <duration>3.0</duration>
          <flyToMode>smooth</flyToMode>
          <LookAt>
            <longitude>174.064</longitude>
            <latitude>-39.321</latitude>
            <altitude>0</altitude>
            <heading>-48.463</heading>
            <tilt>67.946</tilt>
            <range>4202.579</range>
            <altitudeMode>relativeToGround</altitudeMode>
          </LookAt>
        </FlyTo>
        <Wait>
          <duration>6.0</duration>
        </Wait>
      </Playlist>
    </Tour>
    <Folder>
      <name>Points and polygons</name>
      <Style id="pushpin">
        <IconStyle>
          <Icon>
            <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>
```

```
          </Icon>
        </IconStyle>
      </Style>
      <Placemark id="mountainpin1">
        <name>New Zealand's Southern Alps</name>
        <styleUrl>#pushpin</styleUrl>
        <Point>
          <coordinates>170.144,-43.605,0</coordinates>
        </Point>
      </Placemark>
      <Placemark id="polygon1">
        <name>Polygon</name>
        <Polygon>
          <tessellate>1</tessellate>
          <outerBoundaryIs>
            <LinearRing>
              <coordinates> 175.365,-36.522,0 175.366,-36.530,0 175.369,-
36.529,0 175.366,-36.521,0
                175.365,-36.522,0 </coordinates>
            </LinearRing>
          </outerBoundaryIs>
        </Polygon>
      </Placemark>
    </Folder>
  </Document>
</kml>
```

9.24 kml:Playlist

9.24.1 Structure

```
<kml:Playlist
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveGroup>...</kml:AbstractTourPrimitiveGroup> [0..*]
   <kml:PlaylistSimpleExtensionGroup>...</kml:PlaylistSimpleExtensionGroup>
[0..*]
   <kml:PlaylistObjectExtensionGroup>...</kml:PlaylistObjectExtensionGroup>
[0..*]
</kml:Playlist>
```

9.24.2 Description

The `kml:Playlist` element contains any number of tour primitives, i.e. elements that substitute for *`kml:AbstractTourPrimitiveGroup`*. There can be at most one `kml:Playlist` element contained within a `kml:Tour` element.

9.24.3 Content

9.24.3.1 kml:AbstractTourPrimitiveGroup

An abstract placeholder for the following instantiable elements: `kml:FlyTo`, `kml:AnimatedUpdate`, `kml:TourControl`, `kml:Wait`, and `kml:SoundCue`.

See 9.25 kml:AbstractTourPrimitiveGroup.

9.24.3.2 kml:PlaylistSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.24.3.3 kml:PlaylistObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.25 kml:AbstractTourPrimitiveGroup

9.25.1 Structure

```
<kml:AbstractTourPrimitiveGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
  <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
   </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveObjectExtensionGroup>...
   </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
</kml:AbstractTourPrimitiveGroup>
```

9.25.2 Description

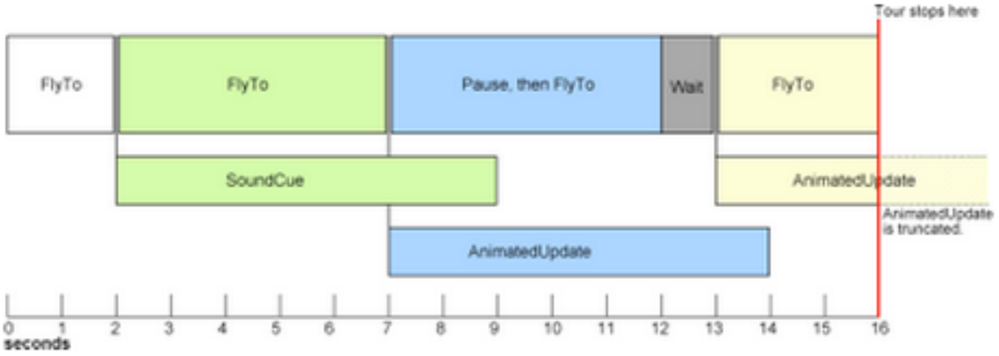
The `kml:AbstractTourPrimitiveGroup` is an abstract placeholder for the following instantiable elements: `kml:FlyTo`, `kml:AnimatedUpdate`, `kml:TourControl`, `kml:Wait`, and `kml:SoundCue`. Some of these primitives are played series and some can be played in parallel with others.

The primitives: `kml:FlyTo`, `kml:Wait`, and `kml:TourControl` are played sequentially, and must be fully complete before the playlist moves on to the next primitive. So only one of these series-type primitives can take place at a time – no new `kml:SoundCue` or `kml:AnimatedUpdate` primitives can be started while a series-type primitive is taking place.

The primitives: `kml:AnimatedUpdate` and `kml:SoundCue` can be played in parallel with other primitives**.** These primitives are cued when they appear in the playlist. Once they are started, the playlist immediately moves on to the next primitive, without waiting for the `kml:SoundCue` or `kml:AnimatedUpdate` to complete. This means that:

- any number of sound cues and/or animated updates can take place at the same time, by cueing them one after the other in the playlist
- sound cues and/or animated updates can continue while series-type primitives are playing (though they can not be *started* while series-type primitives are playing)

The diagram below illustrates this behavior.



The diagram shows that each `kml:FlyTo` takes a specific amount of time to complete, which is specified by its `kml:duration` element. After the duration has elapsed, the tour proceeds to the next primitive.

On the other hand, the `kml:SoundCue`, takes place in parallel. The next primitive (the second `kml:FlyTo`) in the playlist is executed immediately after the sound track begins playing. The sound track continues until the end of the file is reached, or the end of the tour, whichever occurs first. If the tour ends before the sound file does, the sound track is truncated. Any number of sound tracks can be played at any given time, as each is loaded into its own timeline track.

Animated updates behave in a similar way. If a placemark's icon is set to increase in size over a 5 second duration, it does so while the subsequent primitives in the playlist take place.

9.25.3 Content

9.25.3.1 kml:AbstractTourPrimitiveSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.25.3.2 kml: AbstractTourPrimitiveObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.25.4 Example

```
<Tour xmlns="http://www.opengis.net/kml/2.2">
  <Playlist>
    <FlyTo>
      <duration>5</duration>
      <flyToMode>bounce</flyToMode>
      <LookAt>
        <longitude>-79.387</longitude>
        <latitude>43.643</latitude>
        <altitude>0</altitude>
        <heading>-172.3</heading>
        <tilt>10</tilt>
        <range>1200</range>
        <altitudeMode>relativeToGround</altitudeMode>
      </LookAt>
    </FlyTo>
    <TourControl>
      <playMode>pause</playMode>
    </TourControl>
    <SoundCue>
      <href> http://dev.keyhole.com/codesite/cntowerfacts.mp3 </href>
      <!-- this file is ten seconds long -->
    </SoundCue>
    <Wait>
      <duration>10.0</duration>
    </Wait>
  </Playlist>
</Tour>
```

9.26 kml:AnimatedUpdate

9.26.1 Structure

```
<kml:AnimatedUpdate
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
    </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveObjectExtensionGroup>...
    </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
   <kml:duration>...</kml:duration> [0..1]
   <kml:Update>...</kml:Update> [0..1]
   <kml:delayedStart>...</kml:delayedStart> [0..1]
   <kml:AnimatedUpdateSimpleExtensionGroup>...
    </kml:AnimatedUpdateSimpleExtensionGroup> [0..*]
   <kml:AnimatedUpdateObjectExtensionGroup>...
    </kml:AnimatedUpdateObjectExtensionGroup> [0..*]
</kml:AnimatedUpdate>
```

9.26.2 Description

kml:AnimatedUpdate controls changes to KML Features (e.g. balloon open/close) during a tour using kml:Update.

9.26.3 Content

kml:AnimatedUpdate should contain a kml:duration value to specify the length of time in seconds over which the update takes place. Integer, float, and color values are animated by interpolating from the original to the new value over the duration; boolean, string, and other values not amenable to interpolation, are updated at the end of the duration.

9.26.3.1 kml:duration

9.26.3.1.1 Description

Specifies the length of time, in seconds, over which the update takes place.

9.26.3.1.2 Content

Type:	xsd:double
Default Value:	0.0

9.26.3.2 kml:Update

See section 13.5 kml:Update. If the objects to be updated all occur in the same document as the parent `kml:AnimatedUpdate` element then the `kml:Update/kml:targetHref` element may be empty.

9.26.3.3 kml:delayedStart

9.26.3.3.1 Description

Specifies the number of seconds to wait (after the inline start position) before applying the update.

9.26.3.3.2 Content

Type:	xsd:double
Default Value:	0.0

9.26.3.4 kml:AnimatedUpdateSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.26.3.5 kml:AnimatedUpdateObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.26.4 Example

The following example demonstrates a change in icon size and is animated for a duration of 5 seconds.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>AnimatedUpdate example</name>
    <Style id="pushpin">
      <IconStyle id="mystyle">
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-
pushpin.png</href>
          <scale>1.0</scale>
        </Icon>
      </IconStyle>
    </Style>
    <Placemark id="mountainpin1">
      <name>Pin on a mountaintop</name>
      <styleUrl>#pushpin</styleUrl>
      <Point>
        <coordinates>170.1435558771009,-43.60505741890396,0</coordinates>
      </Point>
    </Placemark>
    <Tour>
      <name>Play me!</name>
      <Playlist>
        <FlyTo>
          <flyToMode>bounce</flyToMode>
          <duration>3</duration>
          <Camera>
            <longitude>170.157</longitude>
            <latitude>-43.671</latitude>
            <altitude>9700</altitude>
            <heading>-6.333</heading>
            <tilt>33.5</tilt>
          </Camera>
        </FlyTo>
        <AnimatedUpdate>
          <duration>5</duration>
          <Update>
            <targetHref/>
            <Change>
              <IconStyle targetId="mystyle">
                <scale>10.0</scale>
              </IconStyle>
            </Change>
          </Update>
        </AnimatedUpdate>
        <Wait>
          <duration>5</duration>
        </Wait>
      </Playlist>
    </Tour>
  </Document>
</kml>
```

9.27 kml:FlyTo

9.27.1 Structure

```
<kml:FlyTo
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
    </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveObjectExtensionGroup>...
    </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
   <kml:duration>...</kml:duration> [0..1]
   <kml:abstractFlyToMode>...</kml:abstractFlyToMode> [0..1]
   <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
   <kml:FlyToSimpleExtensionGroup>...</kml:FlyToSimpleExtensionGroup> [0..*]
   <kml:FlyToObjectExtensionGroup>...</kml:FlyToObjectExtensionGroup> [0..*]
</kml:FlyTo>
```

9.27.2 Description

`kml:FlyTo` specifies a point in space to which the browser will fly during a tour. It must contain an abstract view (e.g. `kml:LookAt` or `kml:Camera`), and should contain `kml:duration` and `kml:flyToMode` elements, which specify the time it takes to fly to the specified point and the method of flight, respectively.

9.27.3 Content

9.27.3.1 kml:duration

Specifies the length of time, in seconds, over which the flight takes place.

See 9.26.3.1 kml:duration.

9.27.3.2 kml:abstractFlyToMode

9.27.3.2.1 Description

An abstract placeholder for `kml:flyToMode` and an extension point for fly-to mode enumerations. See also 9.28 kml:flyToMode and 6.7.2.1 Simple Element Substitution.

9.27.3.2.2 Content

Type:	kml:enumBaseType
Default Value:	none

9.27.3.3 kml:AbstractViewGroup

An abstract placeholder for a viewpoint (`kml:Camera` or `kml:LookAt`).

See 14.1 AbstractViewGroup.

9.27.3.4 kml:FlyToSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.27.3.5 kml:FlyToObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.27.4 Example

```
<FlyTo>
  <duration>2.55</kml:duration>
  <flyToMode>smooth</kml:flyToMode>
  <Camera>
    <longitude>-113.084448</longitude>
    <latitude>36.567081</latitude>
    <altitude>41277.571403</altitude>
    <heading>116.150227</heading>
    <altitudeMode>absolute</altitudeMode>
  </Camera>
</FlyTo>
```

9.28 kml:flyToMode

9.28.1 Structure

```
<kml:flyToMode>...</kml:flyToMode>
```

9.28.2 Description

Specifies the mode of flight using an enumerated value; **smooth** or **bounce**:

1. **smooth** mode allows for an unbroken flight from point to point. An unbroken series of smooth FlyTos will begin and end at zero velocity, and will not slow at each intermediate point. A sequence of smooth FlyTos will be broken by either of the following elements:

   - `<kml:flyToMode>bounce</kml:flyToMode>`
   - `kml:Wait`

   This means that the velocity will approach zero immediately preceding either of the above elements during a smooth FlyTo. A sequence of smooth FlyTos is not broken by `kml:AnimatedUpdate` elements.

2. **bounce** mode alwaysbegins and ends at zero velocity.

9.28.3 Content

Type:	kml:flyToModeEnumType
Default Value:	bounce

9.29 kml:SoundCue

9.29.1 Structure

```
<kml:SoundCue
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
  <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
   </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveObjectExtensionGroup>...
   </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
  <kml:href>...</kml:href> [0..1]
  <kml:delayedStart>...</kml:delayedStart> [0..1]
  <kml:SoundCueSimpleExtensionGroup>...
   </kml:SoundCueSimpleExtensionGroup> [0..*]
  <kml:SoundCueObjectExtensionGroup>...
   </kml:SoundCueObjectExtensionGroup> [0..*]
</kml:SoundCue>
```

9.29.2 Description

Contains a `kml:href` element to reference a sound track. The `kml:SoundCue` element does not contain a duration. The sound file plays in parallel to the rest of the tour, meaning that the next tour primitive takes place immediately after the `kml:SoundCue` tour primitive is reached. If another sound file is cued before the first has finished playing, the files are mixed. The `kml:delayedStart` element specifies to delay the start of the sound for a given number of seconds before playing the file.

9.29.3 Content

9.29.3.1 kml:href

Specfies a URI reference to a sound track (e.g. MP3, M4A, or AAC format).

See 12.13.3.1 kml:href.

9.29.3.2 kml:delayedStart

Specifies the number of seconds to wait before playing the sound track.

See 9.26.3.3 kml:delayedStart.

9.29.3.3 kml:SoundCueSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.29.3.4 kml:SoundCueObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.29.4 Example

```
<Tour>
  <Playlist>
    <SoundCue>
      <!-- 15 second audio clip -->
      <href>http://www.example.com/audio/trumpets.mp3</href>
      <delayedStart>1</kml:delayedStart>
    </SoundCue>
    <FlyTo>
      <duration>5</duration>
      <flyToMode>bounce</flyToMode>
      <LookAt>
        <longitude>-79.387</longitude>
        <latitude>43.643</latitude>
        <altitude>0</altitude>
        <range>1200</range>
        <tilt>10</tilt>
        <heading>-172.3</heading>
        <altitudeMode>relativeToGround</altitudeMode>
      </LookAt>
    </FlyTo>
    <SoundCue>
      <href> http://dev.keyhole.com/codesite/cntowerfacts.mp3 </href>
      <!-- 10 second audio clip -->
    </SoundCue>
    <Wait>
      <duration>10</duration>
      <!-- continues the tour for 10 seconds while audio clip plays -->
    </Wait>
  </Playlist>
</Tour>
```

9.30 kml:TourControl

9.30.1 Structure

```
<kml:TourControl
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
  <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
   </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
  <kml:AbstractTourPrimitiveObjectExtensionGroup>...
   </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
  <kml:abstractPlayMode>...</kml:abstractPlayMode> [0..1]
  <kml:TourControlSimpleExtensionGroup>...
   </kml:TourControlSimpleExtensionGroup> [0..*]
  <kml:TourControlObjectExtensionGroup>...
   </kml:TourControlObjectExtensionGroup> [0..*]
</kml:TourControl>
```

9.30.2 Description

`kml:TourControl` allows a `kml:playMode` element, enabling the tour to be paused until a user takes action to continue the tour.

9.30.3 Content

9.30.3.1 kml:abstractPlayMode

9.30.3.1.1 Description

An abstract placeholder for `kml:playMode` and an extension point for play mode enumerations. See also 9.31 kml:playMode and 6.7.2.1 Simple Element Substitution.

9.30.3.1.2 Content

Type:	kml:enumBaseType
Default Value:	none

9.30.3.2 kml:TourControlSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.30.3.3 kml:TourControlObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

9.30.4 Example

```
<TourControl id="TC001">
  <playMode>pause</playMode>
</TourControl>
```

9.31 kml:playMode

9.31.1 Structure

```
<kml:playMode>...</kml:playMode>
```

9.31.2 Description

Specifies the play mode using an enumerated value.

9.31.3 Content

Type:	kml:playModeEnumType
Default Value:	pause

9.32 kml:Wait

9.32.1 Structure

```
<kml:Wait
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveSimpleExtensionGroup>...
    </kml:AbstractTourPrimitiveSimpleExtensionGroup> [0..*]
   <kml:AbstractTourPrimitiveObjectExtensionGroup>...
    </kml:AbstractTourPrimitiveObjectExtensionGroup> [0..*]
   <kml:duration>...</kml:duration> [0..1]
   <kml:WaitSimpleExtensionGroup>...</kml:WaitSimpleExtensionGroup> [0..*]
   <kml:WaitObjectExtensionGroup>...</kml:WaitObjectExtensionGroup> [0..*]
</kml:Wait>
```

9.32.2 Description

The `kml:Wait` element holds the camera still, at the last-defined abstract view, for the number of seconds specified in its `kml:duration` element before playing the next tour primitive. Note that `kml:Wait` does not pause the tour timeline – sound files and animated updates will continue to play while the camera is waiting.

9.32.3 Content

9.32.3.1 kml:duration

Specifies the length of time, in seconds, to wait while holding the camera still at the last defined abstract view.

See 9.26.3.1 kml:duration.

9.32.3.2 kml:WaitSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

9.32.3.3 kml:WaitObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10. Geometries

10.1 kml:AbstractGeometryGroup

10.1.1 Structure

```
<kml:AbstractGeometryGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractGeometrySimpleExtensionGroup>...
    </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
   <kml:AbstractGeometryObjectExtensionGroup>...
    </kml:AbstractGeometryObjectExtensionGroup> [0..*]
</kml:AbstractGeometryGroup>
```

10.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:MultiGeometry
- kml:LineString
- kml:Polygon
- kml:Point
- kml:LinearRing
- kml:Model

10.1.3 Content

10.1.3.1 kml:AbstractGeometrySimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.1.3.2 kml:AbstractGeometryObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.2 kml:MultiGeometry

10.2.1 Structure

```
<kml:MultiGeometry
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:AbstractGeometryGroup>...</kml:AbstractGeometryGroup> [0..*]
    <kml:MultiGeometrySimpleExtensionGroup>...
     </kml:MultiGeometrySimpleExtensionGroup> [0..*]
    <kml:MultiGeometryObjectExtensionGroup>...
     </kml:MultiGeometryObjectExtensionGroup> [0..*]
</kml:MultiGeometry>
```

10.2.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

A container for zero or more geometry elements associated with the same KML feature.

A `kml:MultiGeometry` element should contain more than one *kml:AbstractGeometryGroup* element outside of an update context, that is when not a descendant of `kml:Update`.

10.2.3 Content

10.2.3.1 kml:AbstractGeometryGroup

See 10.1 kml:AbstractGeometryGroup.

10.2.3.2 kml:AbstractGeometrySimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.2.3.3 kml:AbstractGeometryObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.2.4 Example

```
<Placemark>
  <name>SF Marina Harbor Master</name>
  <visibility>0</visibility>
  <MultiGeometry>
    <LineString>
      <!-- north wall -->
      <coordinates>
        -122.4425587930444,37.80666418607323,0
        -122.4428379594768,37.80663578323093,0
      </coordinates>
    </LineString>
    <LineString>
      <!-- south wall -->
      <coordinates>
        -122.4425509770566,37.80662588061205,0
        -122.4428340530617,37.8065999493009,0
      </coordinates>
    </LineString>
  </MultiGeometry>
</Placemark>
```

10.3 kml:Point

10.3.1 Structure

```
<kml:Point
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:extrude>...</kml:extrude> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:coordinates>...</kml:coordinates> [0..1]
    <kml:PointSimpleExtensionGroup>...</kml:PointSimpleExtensionGroup> [0..*]
    <kml:PointObjectExtensionGroup>...</kml:PointObjectExtensionGroup> [0..*]
</kml:Point>
```

10.3.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

A geographic location defined by a single geodetic longitude, geodetic latitude, and (optional) altitude coordinate tuple.

10.3.3 Content

10.3.3.1 kml:extrude

Specifies whether to connect the point to the ground with a line when displayed in the geographic view.

See 10.4 kml:extrude.

10.3.3.2 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.3.3.3 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.3.3.4 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.3.3.5 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.3.3.6 kml:coordinates

10.3.3.6.1 Description

A single coordinate tuple.

10.3.3.6.2 Content

Type:	kml:coordinatesType
Default Value:	none

10.3.3.7 kml:PointSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.3.3.8 kml:PointObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.3.4 Example

```
<Point>
  <coordinates>-90.86948943473118,48.25450093195546</coordinates>
</Point>
```

10.4 kml:extrude

10.4.1 Structure

```
<kml:extrude>...</kml:extrude>
```

10.4.2 Description

Specifies whether to connect a geometry to the ground. To extrude a geometry, the `kml:altitudeMode` shall be either **relativeToGround** or **absolute**, and the altitude component within the `kml:coordinates` element should be greater than 0 (that is, in the air). The geometry is extruded toward the Earth's center of mass.

See also 6.3 Geometry Interpolation for 3D Earth Browsers.

10.4.3 Content

Type:	xsd:boolean
Default Value:	0 or false

10.5 kml:LinearRing

10.5.1 Structure

```
<kml:LinearRing
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
     <kml:extrude>...</kml:extrude> [0..1]
     <kml:tessellate>...</kml:tessellate> [0..1]
     <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
     <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
     <kml:AltitudeModeSimpleExtensionGroup>...
      </kml:AltitudeModeSimpleExtensionGroup> [0..1]
     <kml:AltitudeModeObjectExtensionGroup>...
      </kml:AltitudeModeObjectExtensionGroup> [0..1]
     <kml:coordinates>...</kml:coordinates> [0..1]
     <kml:altitudeOffset>...</kml:altitudeOffset> [0..1]
     <kml:LinearRingSimpleExtensionGroup>...
      </kml:LinearRingSimpleExtensionGroup> [0..*]
     <kml:LinearRingObjectExtensionGroup>...
      </kml:LinearRingObjectExtensionGroup> [0..*]
</kml:LinearRing>
```

10.5.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

Defines a closed line string that should not cross itself.

kml:extrude, kml:tessellate, kml:altitudeMode and kml:seaFloorAltitudeMode elements should not be specified when kml:LinearRing is used to define a boundary for a kml:Polygon.

10.5.3 Content

10.5.3.1 kml:extrude

See 10.4 kml:extrude.

10.5.3.2 kml:tessellate

See 10.6 kml:tessellate.

10.5.3.3 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.5.3.4 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.5.3.5 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.5.3.6 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.5.3.7 kml:coordinates

10.5.3.7.1 Description

A list of four or more coordinate tuples where the first and last coordinate tuples must be the same.

10.5.3.7.2 Content

Type:	kml:coordinatesType
Default Value:	none

10.5.3.8 kml:altitudeOffset

10.5.3.8.1 Description

Modifies how the altitude values are rendered. The value of `kml:altitudeOffset` shifts the entire geometry up or down as a unit. Units are in meters.

10.5.3.8.2 Content

Type:	xsd:double
Default Value:	0.0

10.5.3.9 kml:LinearRingSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.5.3.10 kml:LinearRingObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.5.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
  <name>LinearRing.kml</name>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          -122.365662,37.826988,0
          -122.365202,37.826302,0
          -122.364581,37.82655,0
          -122.365038,37.827237,0
          -122.365662,37.826988,0
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>
</kml>
```

10.6 kml:tessellate

10.6.1 Structure

```
<kml:tessellate>...</kml:tessellate>
```

10.6.2 Description

Specifies whether to drape a geometry over the terrain. A value of 1 or true specifies to drape; a value of 0 or false specifies not to drape. To enable tessellation, the value for `kml:altitudeMode` shall be **clampToGround**.

See also 6.3 Geometry Interpolation for 3D Earth Browsers.

10.6.3 Content

Type:	xsd:boolean
Default Value:	0 or false

10.7 kml:LineString

10.7.1 Structure

```
<kml:LineString
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:extrude>...</kml:extrude> [0..1]
    <kml:tessellate>...</kml:tessellate> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:coordinates>...</kml:coordinates> [0..1]
    <kml:altitudeOffset>...</kml:altitudeOffset> [0..1]
    <kml:LineStringSimpleExtensionGroup>...
     </kml:LineStringSimpleExtensionGroup> [0..*]
    <kml:LineStringObjectExtensionGroup>...
     </kml:LineStringObjectExtensionGroup> [0..*]
</kml:LineString>
```

10.7.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

Defines a list of one or more contiguous line segments.

10.7.3 Content

10.7.3.1 kml:extrude

Specifies whether to extend the `kml:LineString` to the ground when displayed in the geographic view. When a `kml:LineString` is extruded, the points of each line segment are extended to the terrain toward the Earth's center of mass, forming a polygon resembling a wall or fence.

See 10.4 kml:extrude.

10.7.3.2 kml:tessellate

See 10.6 kml:tessellate.

10.7.3.3 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.7.3.4 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.7.3.5 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.7.3.6 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.7.3.7 kml:coordinates

10.7.3.7.1 Description

Two or more coordinate tuples.

10.7.3.7.2 Content

Type:	kml:coordinatesType
Default Value:	none

10.7.3.8 kml:altitudeOffset

See 10.5.3.8 kml:altitudeOffset.

10.7.3.9 kml:LineStringSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.7.3.10 kml:LineStringObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.7.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>LineString.kml</name>
  <open>1</open>
  <LookAt>
    <longitude>-122.36415</longitude>
    <latitude>37.824553</latitude>
    <altitude>0</altitude>
    <range>150</range>
    <tilt>50</tilt>
    <heading>0</heading>
  </LookAt>
  <Placemark>
    <name>unextruded</name>
    <LineString>
      <extrude>0</extrude>
      <tessellate>1</tessellate>
      <coordinates>
        -122.364383,37.824664,0 -122.364152,37.824322,0
      </coordinates>
    </LineString>
  </Placemark>
  <Placemark>
    <name>extruded</name>
    <LineString>
      <extrude>1</extrude>
      <tessellate>1</tessellate>
      <altitudeMode>relativeToGround</altitudeMode>
      <coordinates>
        -122.364167,37.824787,50 -122.363917,37.824423,50
      </coordinates>
    </LineString>
  </Placemark>
</Document>
</kml>
```

10.8 kml:Polygon

10.8.1 Structure

```
<kml:Polygon
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:extrude>...</kml:extrude> [0..1]
    <kml:tessellate>...</kml:tessellate> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:outerBoundaryIs>...</kml:outerBoundaryIs> [0..1]
    <kml:innerBoundaryIs>...</kml:innerBoundaryIs> [0..*]
    <kml:PolygonSimpleExtensionGroup>...
     </kml:PolygonSimpleExtensionGroup> [0..*]
    <kml:PolygonObjectExtensionGroup>...
     </kml:PolygonObjectExtensionGroup> [0..*]
</kml:Polygon>
```

10.8.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

A kml:Polygon shall have an outer boundary ring outside of an update context, that is when not a descendant of kml:Update. It may have 0 or more inner boundary rings. Each ring is defined by a kml:LinearRing. It is advised that the rings not cross each other.

10.8.3 Content

10.8.3.1 kml:extrude

10.8.3.1.1 Description

Specifies whether to connect the kml:Polygon to the ground when displayed in the geographic view.

When a kml:Polygon is extruded, each boundary point is extended to the terrain toward the earth's center of mass, which gives the appearance of a building or a box.

Only the kml:Polygon boundary is extruded, not the kml:Polygon interior (for example, a rectangle turns into a box with five faces).

See 10.4 kml:extrude.

10.8.3.2 kml:tessellate

See 10.6 kml:tessellate.

10.8.3.3 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.8.3.4 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.8.3.5 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.8.3.6 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.8.3.7 kml:outerBoundaryIs

Specifies the exterior boundary of the `kml:Polygon` defined by a `kml:LinearRing` element.

See 10.5 kml:LinearRing.

10.8.3.8 kml:innerBoundaryIs

Specifies an inner boundary of the `kml:Polygon` defined by a `kml:LinearRing` element.

See 10.5 kml:LinearRing.

10.8.3.9 kml:PolygonSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.8.3.10 kml:PolygonObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.8.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>Polygon.kml</name>
  <open>0</open>
  <Placemark>
    <name>hollow box</name>
    <Polygon>
      <extrude>1</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.366278,37.818844,30
            -122.365248,37.819267,30
            -122.365640,37.819861,30
            -122.366669,37.819429,30
            -122.366278,37.818844,30
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.366212,37.818977,30
            -122.365424,37.819294,30
            -122.365704,37.819731,30
            -122.366488,37.819402,30
            -122.366212,37.818977,30
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
    </Polygon>
  </Placemark>
</Document>
</kml>
```

10.9 kml:Model

10.9.1 Structure

```
<kml:Model
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractGeometrySimpleExtensionGroup>...
     </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
    <kml:AbstractGeometryObjectExtensionGroup>...
     </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:Location>...</kml:Location> [0..1]
    <kml:Orientation>...</kml:Orientation> [0..1]
    <kml:Scale>...</kml:Scale> [0..1]
    <kml:Link>...</kml:Link> [0..1]
    <kml:ResourceMap>...</kml:ResourceMap> [0..1]
    <kml:ModelSimpleExtensionGroup>...
     </kml:ModelSimpleExtensionGroup> [0..*]
    <kml:ModelObjectExtensionGroup>...
     </kml:ModelObjectExtensionGroup> [0..*]
</kml:Model>
```

10.9.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractGeometryGroup*

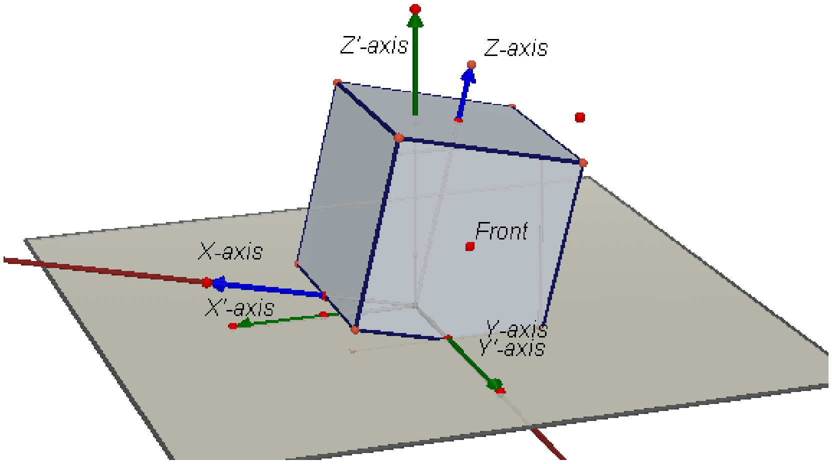
Specifies the location and orientation of a textured 3D object resource. The structure and appearance of the textured 3D object are not defined in this specification.

A local earth-fixed frame (X’-Y’-Z’) is defined as follows.

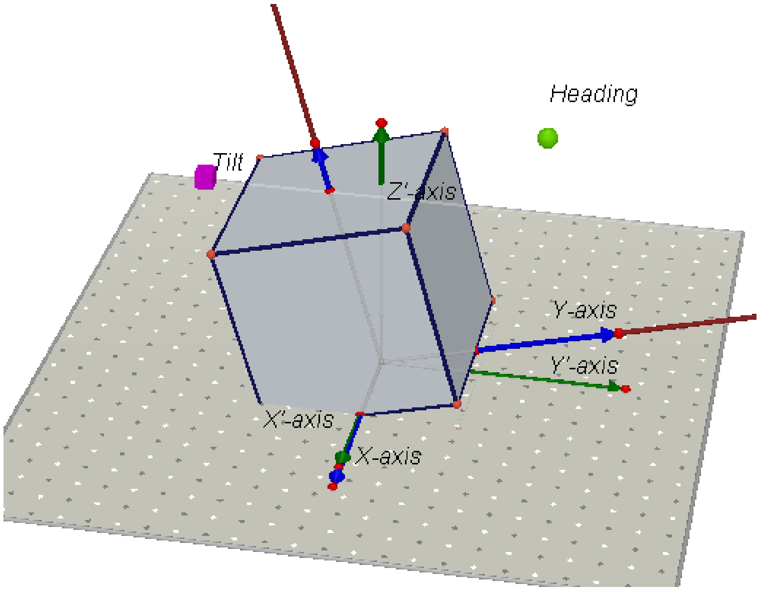
The Z'-axis is defined by the local surface normal (local to the tangent plane at the location point), the Y'-axis is defined by the intersection of the meridian plane defined by the longitude of the location point, and the tangent plane, and the X'-axis is chosen so that X'-Y'-Z' form a right handed orthogonal frame.

A model-fixed frame (X-Y-Z) is defined as follows. The Y-axis is +ve through the front of the model. The Z-axis is +ve through the top of the model and the X-axis is chosen so that X-Y-Z forms a right handed orthogonal frame.

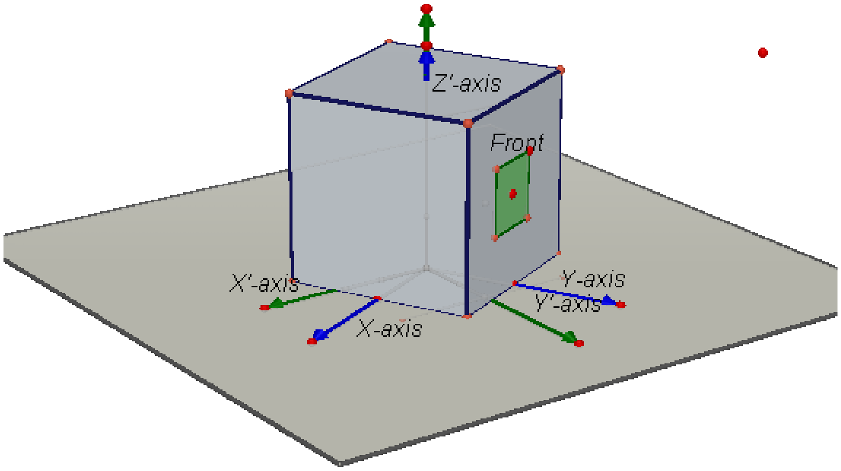
The orientation of the model is then defined by the orientation of the model-fixed frame (X-Y-Z) relative to the local earth-fixed frame (X’-Y’-Z’) as follows:



Roll – rotation about the model Y-axis



Tilt – rotation about the model X-axis



Heading – Rotation about the model Z-axis

`kml:Model`:

□ shall contain `kml:Link` and `kml:Location` elements;

□ shall contain a `kml:ResourceMap` element that has a `kml:Alias` element for each texture file, if the target resource has texture files;

outside of an update context, that is when not a descendant of `kml:Update`.

10.9.3 Content

10.9.3.1 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.9.3.2 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.9.3.3 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.9.3.4 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.9.3.5 kml:Location

10.9.3.5.1 Description

Specifies the coordinates of the `kml:Model's` origin.

See 10.10 kml:Location.

10.9.3.6 kml:Orientation

10.9.3.6.1 Description

Specifies the orientation of the model coordinate axes relative to a local earth-fixed reference frame.

See 10.11 kml:Orientation.

10.9.3.7 kml:Scale

10.9.3.7.1 Description

Scales a model along the *x*, *y*, and *z* axes in the model's coordinate space.

See 10.12 kml:Scale.

10.9.3.8 kml:Link

10.9.3.8.1 Description

Specifies the location of a textured 3D object resource such as a COLLADA file.

See 13.1 kml:Link, kml:Icon (kml:LinkType).

10.9.3.9 kml:ResourceMap

10.9.3.9.1 Description

Specifies 0 or more `kml:Alias` elements, each of which is a mapping for the texture file path from the original textured 3D object file to the KML or KMZ resource that contains the `kml:Model`.

See 10.13 kml:ResourceMap.

10.9.3.10 kml:ModelSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.9.3.11 kml:ModelObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.9.4 Example

```
<Model id="khModel543">
  <altitudeMode>relativeToGround</altitudeMode>
  <Location>
    <longitude>39.55375305703105</longitude>
    <latitude>-118.9813220168456</latitude>
    <altitude>1223</altitude>
  </Location>
  <Orientation>
    <heading>45.0</heading>
    <tilt>10.0</tilt>
    <roll>0.0</roll>
  </Orientation>
  <Scale>
    <x>1.0</x>
    <y>1.0</y>
    <z>1.0</z>
  </Scale>
  <Link>
    <href>house.dae</href>
  </Link>
  <ResourceMap>
    <Alias>
      <targetHref>../files/CU-Macky---Center-StairsnoCulling.jpg</targetHref>
      <sourceHref>CU-Macky---Center-StairsnoCulling.jpg</sourceHref>
    </Alias>
    <Alias>
      <targetHref>../files/CU-Macky-4sideturretnoCulling.jpg</targetHref>
      <sourceHref>CU-Macky-4sideturretnoCulling.jpg</sourceHref>
    </Alias>
    <Alias>
      <targetHref>../files/CU-Macky-Back-NorthnoCulling.jpg</targetHref>
      <sourceHref>CU-Macky-Back-NorthnoCulling.jpg</sourceHref>
    </Alias>
  </ResourceMap>
</Model>
```

10.10 kml:Location

10.10.1 Structure

```
<kml:Location
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:longitude>...</kml:longitude> [0..1]
    <kml:latitude>...</kml:latitude> [0..1]
    <kml:altitude>...</kml:altitude> [0..1]
    <kml:LocationSimpleExtensionGroup>...</kml:LocationSimpleExtensionGroup>
[0..*]
    <kml:LocationObjectExtensionGroup>...</kml:LocationObjectExtensionGroup>
[0..*]
</kml:Location>
```

10.10.2 Description

This element can be used wherever the following element is referenced:

- *`kml:AbstractObjectGroup`*

Specifies the coordinates of a location.

A `kml:Location` element shall contain the `kml:longitude`, `kml:latitude` child elements outside of an update context, that is when not a descendant of `kml:Update`.

10.10.3 Content

10.10.3.1 kml:latitude

10.10.3.1.1 Description

Geodetic latitude of origin in decimal degrees.

10.10.3.1.2 Content

Type:	kml:angle90Type
Default Value:	0.0

10.10.3.2 kml:longitude

10.10.3.2.1 Description

Geodetic longitude of origin in decimal degrees.

10.10.3.2.2 Content

Type:	kml:angle180Type
Default Value:	0.0

10.10.3.3 kml:altitude

10.10.3.3.1 Description

Altitude of origin measured in meters and interpreted according to `kml:altitudeMode`.

10.10.3.3.2 Content

Type:	xsd:double
Default Value:	0.0

10.10.3.4 kml:LocationSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.10.3.5 kml:LocationObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.10.4 Example

```
<Location>
  <longitude>39.55375305703105</longitude>
  <latitude>-118.9813220168456</latitude>
  <altitude>1223</altitude>
</Location>
```

10.11 kml:Orientation

10.11.1 Structure

```
<kml:Orientation
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:heading>...</kml:heading> [0..1]
    <kml:tilt>...</kml:tilt> [0..1]
    <kml:roll>...</kml:roll> [0..1]
    <kml:OrientationSimpleExtensionGroup>...
     </kml:OrientationSimpleExtensionGroup> [0..*]
    <kml:OrientationObjectExtensionGroup>...
     </kml:OrientationObjectExtensionGroup> [0..*]
</kml:Orientation>
```

10.11.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Specifies the orientation of the model coordinate axes relative to a local earth-fixed reference frame.

kml:Orientation should contain at least one child element outside of an update context, that is when not a descendant of kml:Update. It is advised that kml:heading, kml:tile and kml:roll all be specified.

10.11.3 Content

10.11.3.1 kml:heading

10.11.3.1.1 Description

Rotation about the *z* axis. A value of 0 equals North. A positive rotation is counter clockwise around the positive *z* axis, looking along the *z*-axis away from the origin, and specified in decimal degrees from 0 to ±180.

10.11.3.1.2 Content

Type:	kml:angle360Type
Default Value:	0.0

10.11.3.2 kml:tilt

10.11.3.2.1 Description

Rotation about the *x* axis. A positive rotation is counter clockwise around the positive *x* axis and specified in decimal degrees from 0 to ±180.

10.11.3.2.2 Content

Type:	kml:anglepos180Type
Default Value:	0.0

10.11.3.3 kml:roll

10.11.3.3.1 Description

Rotation about the *y* axis. A positive rotation is counter clockwise around the positive *y* axis and specified in decimal degrees from 0 to ±180.

10.11.3.3.2 Content

Type:	kml:angle180Type
Default Value:	0.0

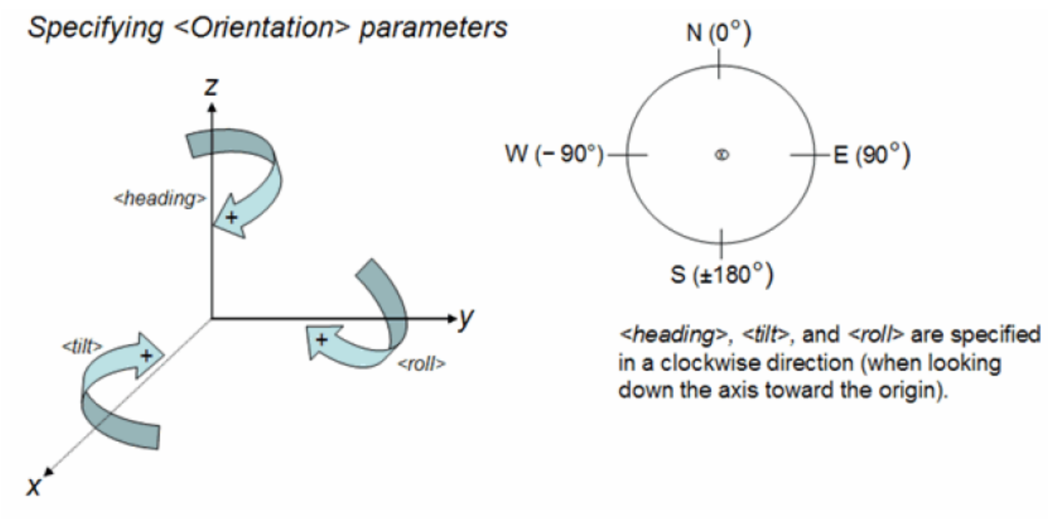
10.11.3.4 kml:OrientationSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.11.3.5 kml:OrientationObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

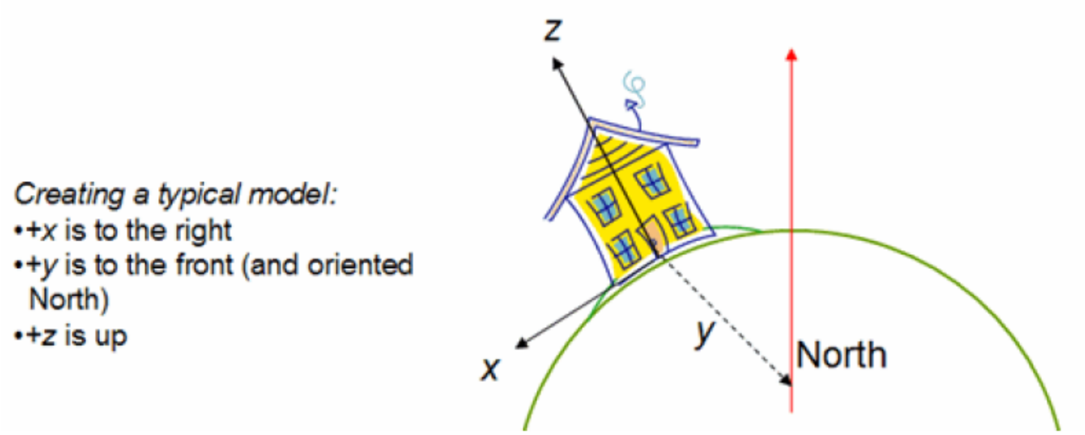
10.11.4 Defining Orientation



Specifying <Orientation> parameters

<heading>, *<tilt>*, and *<roll>* are specified in a clockwise direction (when looking down the axis toward the origin).

This diagram illustrates the initial orientation of a model's axes:


Creating a typical model:
•+x is to the right
•+y is to the front (and oriented North)
•+z is up
z
x
y
North

10.11.5 Example

```
<Orientation>
  <heading>45.0</heading>
  <tilt>10.0</tilt>
  <roll>0.0</roll>
</Orientation>
```

10.12 kml:Scale

10.12.1 Structure

```
<kml:Scale
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:x>...</kml:x> [0..1]
    <kml:y>...</kml:y> [0..1]
    <kml:z>...</kml:z> [0..1]
    <kml:ScaleSimpleExtensionGroup>...</kml:ScaleSimpleExtensionGroup> [0..*]
    <kml:ScaleObjectExtensionGroup>...</kml:ScaleObjectExtensionGroup> [0..*]
</kml:Scale>
```

10.12.2 Description

This element can be used wherever the following element is referenced:

- `kml:AbstractObjectGroup`

Scales a model along the x, y, and z axes in the model's coordinate space.

`kml:Scale` should contain at least one child element outside of an update context, that is when not a descendant of `kml:Update`. It is advised that `kml:x`, `kml:y`, and `kml:z` all be specified.

10.12.3 Content

10.12.3.1 x

10.12.3.1.1 Description

Scale factor along *x* axis.

10.12.3.1.2 Content

Type:	xsd:double
Default Value:	1.0

10.12.3.2 y

10.12.3.2.1 Description

Scale factor along *y* axis.

10.12.3.2.2 Content

Type:	xsd:double
Default Value:	1.0

10.12.3.3 z

10.12.3.3.1 Description

Scale factor along *z* axis.

10.12.3.3.2 Content

Type:	xsd:double
Default Value:	1.0

10.12.3.4 kml:ScaleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.12.3.5 kml:ScaleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.12.4 Example

```
<Scale>
  <x>2.5</x>
  <y>2.5</y>
  <z>3.5</z>
</Scale>
```

10.13 kml:ResourceMap

10.13.1 Structure

```
<kml:ResourceMap
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:Alias>...</kml:Alias> [0..*]
    <kml:ResourceMapSimpleExtensionGroup>...
     </kml:ResourceMapSimpleExtensionGroup> [0..*]
    <kml:ResourceMapObjectExtensionGroup>...
     </kml:ResourceMapObjectExtensionGroup> [0..*]
</kml:ResourceMap>
```

10.13.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Should specify at least one or more kml:Alias elements outside of an update context, that is when not a descendant of kml:Update. Each kml:Alias element is a mapping for the texture file path from the original textured 3D object file to the KML or KMZ resource that contains the kml:Model. This element allows texture files to be moved and renamed without having to update the original textured 3D object file that references those textures. One kml:ResourceMap element can contain multiple mappings from different source textured object files into the same target resource.

Each child kml:Alias element should have a unique child kml:sourceHref element .

10.13.3 Content

10.13.3.1 kml:Alias

See 10.14 kml:Alias.

10.13.3.2 kml:ResourceMapSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.13.3.3 kml:ResourceMapObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.14 kml:Alias

10.14.1 Structure

```
<kml:Alias
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:targetHref>...</kml:targetHref> [0..1]
    <kml:sourceHref>...</kml:sourceHref> [0..1]
    <kml:AliasSimpleExtensionGroup>...</kml:AliasSimpleExtensionGroup> [0..*]
    <kml:AliasObjectExtensionGroup>...</kml:AliasObjectExtensionGroup> [0..*]
</kml:Alias>
```

10.14.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Contains a mapping from `kml:sourceHref` to `kml:targetHref`. Both should be specified ouside of an update context, that is when not a descendant of `kml:Update`.

10.14.3 Content

10.14.3.1 targetHref

10.14.3.1.1 Description

Specifies the textured 3D object file to be fetched by an earth browser. This reference can be a relative reference to an image file within a KMZ file, or it can be an absolute reference to the file (for example, a URL).

10.14.3.1.2 Content

Type:	xsd:anyURI
Default Value:	none

10.14.3.2 sourceHref

10.14.3.2.1 Description

Specifies the path for the texture file within the textured 3D object.

10.14.3.2.2 Content

Type:	xsd:anyURI
Default Value:	none

10.14.3.3 kml:AliasSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.14.3.4 kml:AliasObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.14.4 Example

```
<Alias>
  <targetHref>../images/foo.jpg</targetHref>
  <sourceHref>in-geometry-file/foo.jpg</sourceHref>
</Alias>
```

10.15 kml:Track

10.15.1 Structure

```
<kml:Track
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...
    </kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractGeometrySimpleExtensionGroup>...
    </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
   <kml:AbstractGeometryObjectExtensionGroup>...
    </kml:AbstractGeometryObjectExtensionGroup> [0..*]
   <kml:extrude>...</kml:extrude> [0..1]
   <kml:tessellate>...</kml:tessellate> [0..1]
   <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
   <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
   <kml:AltitudeModeSimpleExtensionGroup>...
    </kml:AltitudeModeSimpleExtensionGroup> [0..1]
   <kml:AltitudeModeObjectExtensionGroup>...
    </kml:AltitudeModeObjectExtensionGroup> [0..1]
   <kml:when>...</kml:when> [0..*]
   <kml:coord>...</kml:coord> [0..*]
   <kml:angles>...</kml:angles> [0..*]
   <kml:Model>...</kml:Model> [0..1]
   <kml:ExtendedData>...</kml:ExtendedData> [0..1]
   <kml:TrackSimpleExtensionGroup>...</kml:TrackSimpleExtensionGroup> [0..*]
   <kml:TrackObjectExtensionGroup>...</kml:TrackObjectExtensionGroup> [0..*]
</kml:Track>
```

10.15.2 Description

A `kml:Track` element captures the path of a moving object over a specified period of time. Either an icon or a model can be used to visually represent the time-stamped positions of the track. The time positions of a `kml:Track` are captured in an array of `kml:when` elements and the space coordinates are captured in an equal-length array of `kml:coord` elements. Additional data arrays of equal-length containing: angle/orientation data (`kml:angles`) or arbitrary data values (`kml:ExtendedData`) may accompany the `kml:when` and `kml:coord` arrays. Note that the time-varying properties of a `kml:Track` (kml:when, kml:coord, kml:angles) may occur in any order. However, these elements must be treated as an implicit, parallel array in which document order is significant. That is, the property values that correspond to the same index are implicitly the fields of a single tuple: (kml:when[i], kml:coord[i], kml:angles[i]).

A kml:Track is essentially a univariate or multivariate time-series (using the terminology of statistics) with either a uniform or non-uniform sampling. If the time values of the kml:when array are at regular intervals, the sampling is uniform, otherwise the sampling is non-uniform. If the kml:when array is accompanied by more than one other data array (e.g. kml:coord, kml:angles, and kml:ExtendedData arrays), the time-series is multivariate, otherwise if accompanied by a single data array (e.g. kml:coord) the time-series is univariate.

> Note: KML versions prior to 2.3 allowed a user to associate a time element with any KML Feature (placemark, ground overlay, etc.). However, only *one* time element could be associated with a given Feature. The kml:Track provides a compact encoding of time series data by associating a single object with multiple time elements. The kml:Track element also provides additional functionality than the earlier mechanism, for example, interpolation of position along the track. With this new functioality, browser implementations can display graphs of elevation and speed (plus additional data, if provided) for the object over time.

10.15.2.1 Missing Data

Missing data values for positions on the track, can be accommodated by empty kml:coord (i.e. <kml:coord></kml:coord>) or kml:angles values. Any element except kml:when can be empty and may be used to represent missing data to balance the size of the kml:when, kml:coord, and kml:angles arrays, since the data array sizes must be equal. Missing numerical data, for example represented by an empty kml:coord or kml:angles, can be estimated by linear interpolation between two well-specified data points nearby. Such interpolation can also be applied to missing numerical data arrays in kml:ExtendedData for a track.

10.15.3 Content

10.15.3.1 kml:extrude

See 10.4 kml:extrude.

10.15.3.2 kml:tessellate

See 10.6 kml:tessellate.

10.15.3.3 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.15.3.4 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.15.3.5 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.15.3.6 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.15.3.7 kml:when

10.15.3.7.1 Description

Specifies a time value that corresponds to a spatial location (specified in by the `kml:coord` element). The number of `kml:when` elements in a `kml:Track` must be equal to the number of `kml:coord` elements (and `kml:angles` elements, if included).

10.15.3.7.2 Content

Type:	kml:dateTimeType
Default Value:	none

10.15.3.8 kml:coord

10.15.3.8.1 Description

Specifies a coordinate value consisting of three single-space delimited floating point values for longitude, latitude, and altitude, with (no comma separators). For example:

```
<kml:coord>-122.207881 37.371915 156.000000</kml:coord>
```

Note that the syntax for the `kml:coord` element is different from `kml:coordinates`, which uses comma separators between the longitude, latitude, and altitude components.

10.15.3.8.2 Content

Type:	xsd:string
Default Value:	none

10.15.3.9 kml:angles

10.15.3.9.1 Description

Specifies additional heading, tilt, and roll values to the icon or model for each time/position within the track. The three floating point values are listed without comma separators and represent degrees of rotation. If `kml:angles` is not specified, then the heading, tilt, and roll values of the object can be interpolated from its track.The number of `kml:angles` elements specified should equal the number of `kml:when` and `kml:coord` elements. An example encoding of `kml:angles` is as follows:

```
<kml:angles>45.54676 66.2342 77.0</kml:angles>
```

10.15.3.9.2 Content

Type:	xsd:string
Default Value:	none

10.15.3.10 kml:Model

If specified, the `kml:Model` replaces the point icon used to indicate the current position on the track. When a `kml:Model` is specified within a `kml:Track`, the child elements of `kml:Model` function as follows:

- The `kml:Location` element is ignored.
- The `kml:altitudeMode` element is ignored.
- The `kml:Orientation` value is combined with the orientation of the track as follows. First, the `kml:Orientation` rotation is applied, which brings the model from its local (x, y, z) coordinate system to a right-side-up, north-facing orientation. Next, a rotation is applied that corresponds to the interpolation of the `kml:angles` values that affect the heading, tilt, and roll of the model as it moves along the track. If no angles are specified, the heading and tilt are inferred from the movement of the model.

See 10.9 kml:Model.

10.15.3.11 kml:ExtendedData

Custom data elements may also be defined in the instance by `kml:Schema` and may be useful to associate with each time/position on a track. For example, tracks representing hikes or bicycle rides could include data for heart rate, cadence, and power, as shown in 10.15.4.2 Example 2 – Extended Data Arrays. In `kml:Schema`, a `kml:SimpleArrayField` is defined for each custom data type. Then, for each data type, a `kml:SimpleArrayData` element in included containing a `kml:value` array corresponding to, and of equal length as, the `kml:when` and `kml:coord` arrays of the track.

See 9.3 kml:ExtendedData.

10.15.3.12 kml:TrackSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.15.3.13 kml:TrackObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.15.4 Examples

The first example shows how to create parallel "arrays" of values for `kml:when` and `kml:coord`. The number of time and position values must be equal.

10.15.4.1 Example 1 – Basic Time and Position Arrays

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder>
    <Placemark>
      <Track>
        <when>2010-05-28T02:02:09Z</when>
        <when>2010-05-28T02:02:35Z</when>
        <when>2010-05-28T02:02:44Z</when>
        <when>2010-05-28T02:02:53Z</when>
        <when>2010-05-28T02:02:54Z</when>
        <when>2010-05-28T02:02:55Z</when>
        <when>2010-05-28T02:02:56Z</when>
        <coord>-122.207881 37.371915 156.000000</coord>
        <coord>-122.205712 37.373288 152.000000</coord>
        <coord>-122.204678 37.373939 147.000000</coord>
        <coord>-122.203572 37.374630 142.199997</coord>
        <coord>-122.203451 37.374706 141.800003</coord>
        <coord>-122.203329 37.374780 141.199997</coord>
        <coord>-122.203207 37.374857 140.199997</coord>
      </Track>
    </Placemark>
  </Folder>
</kml>
```

10.15.4.2 Example 2 – Extended Data Arrays

The boldface type in this example highlights the elements used to define and specify custom data for a bike ride. The custom data fields are internally named `heartrate`, `cadence`, and `power`. The `kml:Schema` element specifies an additional display name (using `kml:displayName`) for each set of values (`Heart Rate`, `Cadence`, and `Power`) and specifies the data type for each new field (`xsd:int`, `xsd:int`, and `xsd:float`, respectively).

This example provides a richer presentation of a track, with custom icons and separate icon and line styles for highlight and normal modes. Note, however, that the example only demonstrates seven sets of data values. The original example includes tens of thousands of values (data courtesy of Sean Broeder). This data was collected with a Garmin Edge 705 with associated heart rate monitor and power meter.

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>GPS record of bike ride</name>

    <!-- Normal track style -->
    <LookAt>
      <TimeSpan>
```

```
        <begin>2010-05-28T02:02:09Z</begin>
        <end>2010-05-28T02:02:56Z</end>
      </TimeSpan>
      <longitude>-122.205544</longitude>
      <latitude>37.373386</latitude>
      <range>1300.000000</range>
    </LookAt>
    <Style id="track_n">
      <IconStyle>
        <scale>.5</scale>
        <Icon>
          <href>http://earth.google.com/images/kml-icons/track-
directional/track-none.png</href>
        </Icon>
      </IconStyle>
      <LabelStyle>
        <scale>0</scale>
      </LabelStyle>

    </Style>
    <!-- Highlighted track style -->
    <Style id="track_h">
      <IconStyle>
        <scale>1.2</scale>
        <Icon>
          <href>http://earth.google.com/images/kml-icons/track-
directional/track-none.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <StyleMap id="track">
      <Pair>
        <key>normal</key>
        <styleUrl>#track_n</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#track_h</styleUrl>
      </Pair>
    </StyleMap>
    <!-- Normal multiTrack style -->
    <Style id="multiTrack_n">
      <IconStyle>
        <Icon>
          <href>http://earth.google.com/images/kml-icons/track-
directional/track-0.png</href>
        </Icon>
      </IconStyle>
      <LineStyle>
        <color>99ffac59</color>
        <width>6</width>
      </LineStyle>

    </Style>
    <!-- Highlighted multiTrack style -->
    <Style id="multiTrack_h">
      <IconStyle>
        <scale>1.2</scale>
        <Icon>
          <href>http://earth.google.com/images/kml-icons/track-
directional/track-0.png</href>
        </Icon>
      </IconStyle>
      <LineStyle>
        <color>99ffac59</color>
        <width>8</width>
      </LineStyle>
    </Style>
    <StyleMap id="multiTrack">
      <Pair>
        <key>normal</key>
        <styleUrl>#multiTrack_n</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#multiTrack_h</styleUrl>
      </Pair>
    </StyleMap>
    <!-- Normal waypoint style -->
    <Style id="waypoint_n">
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/pal4/icon61.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <!-- Highlighted waypoint style -->
    <Style id="waypoint_h">
      <IconStyle>
        <scale>1.2</scale>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/pal4/icon61.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <StyleMap id="waypoint">
      <Pair>
        <key>normal</key>
        <styleUrl>#waypoint_n</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#waypoint_h</styleUrl>
      </Pair>
    </StyleMap>
    <Style id="lineStyle">
      <LineStyle>
        <color>99ffac59</color>
        <width>6</width>
      </LineStyle>
    </Style>
    <Schema id="schema">
      <SimpleArrayField name="heartrate" type="xsd:int">
        <displayName>Heart Rate</displayName>
      </SimpleArrayField>
      <SimpleArrayField name="cadence" type="xsd:int">
        <displayName>Cadence</displayName>
      </SimpleArrayField>
      <SimpleArrayField name="power" type="xsd:float">
        <displayName>Power</displayName>
      </SimpleArrayField>
    </Schema>
    <Folder>
      <name>Tracks</name>
      <Placemark>
        <name>2010-05-28T01:16:35.000Z</name>
        <styleUrl>#multiTrack</styleUrl>
        <Track>
          <when>2010-05-28T02:02:09Z</when>
          <when>2010-05-28T02:02:35Z</when>
          <when>2010-05-28T02:02:44Z</when>
          <when>2010-05-28T02:02:53Z</when>
          <when>2010-05-28T02:02:54Z</when>
          <when>2010-05-28T02:02:55Z</when>
          <when>2010-05-28T02:02:56Z</when>
          <coord>-122.207881 37.371915 156.000000</coord>
          <coord>-122.205712 37.373288 152.000000</coord>
          <coord>-122.204678 37.373939 147.000000</coord>
          <coord>-122.203572 37.374630 142.199997</coord>
          <coord>-122.203451 37.374706 141.800003</coord>
          <coord>-122.203329 37.374780 141.199997</coord>
          <coord>-122.203207 37.374857 140.199997</coord>
          <ExtendedData>
            <SchemaData schemaUrl="#schema">
              <SimpleArrayData name="cadence">
                <value>86</value>
                <value>103</value>
                <value>108</value>
                <value>113</value>
                <value>113</value>
                <value>113</value>
                <value>113</value>
              </SimpleArrayData>
              <SimpleArrayData name="heartrate">
                <value>181</value>
                <value>177</value>
                <value>175</value>
                <value>173</value>
                <value>173</value>
                <value>173</value>
                <value>173</value>
              </SimpleArrayData>
              <SimpleArrayData name="power">
                <value>327.0</value>
                <value>177.0</value>
                <value>179.0</value>
                <value>162.0</value>
                <value>166.0</value>
                <value>177.0</value>
                <value>183.0</value>
              </SimpleArrayData>
            </SchemaData>
          </ExtendedData>
        </Track>
      </Placemark>
    </Folder>
  </Document>
</kml>
```

10.16 kml:MultiTrack

10.16.1 Structure

```
<kml:MultiTrack
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractGeometrySimpleExtensionGroup>...
    </kml:AbstractGeometrySimpleExtensionGroup> [0..*]
   <kml:AbstractGeometryObjectExtensionGroup>...
    </kml:AbstractGeometryObjectExtensionGroup> [0..*]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
   <kml:interpolate>...</kml:interpolate> [0..1]
   <kml:Track>...</kml:Track> [0..*]
   <kml:MultiTrackSimpleExtensionGroup>...
    </kml:MultiTrackSimpleExtensionGroup> [0..*]
   <kml:MultiTrackObjectExtensionGroup>...
    </kml:MultiTrackObjectExtensionGroup> [0..*]
</kml:MultiTrack>
```

10.16.2 Description

A `kml:MultiTrack` element is used to collect multiple tracks into one conceptual unit with an associated icon (or model) that moves along the track. The `kml:MultiTrack` encoding is optimized for multiple tracks for the same real-world object.

10.16.3 Content

10.16.3.1 kml:altitudeMode

See 9.20 kml:altitudeMode.

10.16.3.2 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

10.16.3.3 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.16.3.4 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.16.3.5 kml:interpolate

10.16.3.5.1 Description

Specifies a boolean value indicating whether to interpolate between the tracks in a multi-track or not. If the boolean value is 0 (or '`false`'), then the point icon or model stops at the end of one track and skips to the start of the next one. For example, if a single placemark is needed to represent your path over two separate days, and the GPS unit used to record position was turned off for four hours during this period, a discontinuity between the points where the unit was turned off and then on again will be captured. If the value for `kml:interpolate` is 1 (or '`true`'), the values between the end of the first track and the beginning of the next track are joined by linearly interpolated points to form a single continuous path.

10.16.3.5.2 Content

Type:	xsd:boolean
Default Value:	0 or false

10.16.3.6 kml:Track

Specifies a `kml:Track` member of the `kml:MultiTrack` collection.

See 10.15 kml:Track.

10.16.3.7 kml:MultiTrackSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

10.16.3.8 kml:MultiTrackObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

10.16.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
    <Folder>
      <Placemark>
        <MultiTrack>
          <interpolate>1</interpolate>
          <Track>
            <when>2010-05-28T02:02:09Z</when>
            <when>2010-05-28T02:02:35Z</when>
            <when>2010-05-28T02:02:44Z</when>
            <when>2010-05-28T02:02:53Z</when>
            <when>2010-05-28T02:02:54Z</when>
            <when>2010-05-28T02:02:55Z</when>
            <when>2010-05-28T02:02:56Z</when>
            <coord>-122.203208 37.374857 140.199997</coord>
            <coord>-122.203330 37.374780 141.199997</coord>
            <coord>-122.203452 37.374706 141.800003</coord>
            <coord>-122.203573 37.374630 142.199997</coord>
            <coord>-122.204679 37.373939 147.000000</coord>
            <coord>-122.205713 37.373288 152.000000</coord>
            <coord>-122.207882 37.371915 156.000000</coord>
          </Track>
          <Track>
            <when>2010-05-28T02:02:57Z</when>
            <when>2010-05-28T02:02:58Z</when>
            <when>2010-05-28T02:02:59Z</when>
            <when>2010-05-28T02:03:00Z</when>
            <when>2010-05-28T02:03:01Z</when>
            <when>2010-05-28T02:03:02Z</when>
            <when>2010-05-28T02:03:03Z</when>
            <coord>-122.207881 37.371915 156.000000</coord>
            <coord>-122.205712 37.373288 152.000000</coord>
            <coord>-122.204678 37.373939 147.000000</coord>
            <coord>-122.203572 37.374630 142.199997</coord>
            <coord>-122.203451 37.374706 141.800003</coord>
            <coord>-122.203329 37.374780 141.199997</coord>
            <coord>-122.203207 37.374857 140.199997</coord>
          </Track>
        </MultiTrack>
      </Placemark>
    </Folder>
  </kml>
```

11. Overlays

11.1 kml:AbstractOverlayGroup

11.1.1 Structure

```
<kml:AbstractOverlayGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:drawOrder>...</kml:drawOrder> [0..1]
    <kml:Icon>...</kml:Icon> [0..1]
    <kml:AbstractOverlaySimpleExtensionGroup>...
     </kml:AbstractOverlaySimpleExtensionGroup> [0..*]
    <kml:AbstractOverlayObjectExtensionGroup>...
     </kml:AbstractOverlayObjectExtensionGroup> [0..*]
</kml:AbstractOverlayGroup>
```

11.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractFeatureGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:GroundOverlay
- kml:PhotoOverlay
- kml:ScreenOverlay

Specifies how to display an image specified by `kml:Icon`.

A *kml:AbstractOverlayGroup* element should contain the kml:Icon element outside of an update context, that is when not a descendant of kml:Update.

11.1.3 Content

11.1.3.1 kml:color

11.1.3.1.1 Description

Specifies the color of the graphic element.

11.1.3.1.2 Content

Type:	kml:colorType
Default Value:	ffffffff

11.1.3.2 kml:drawOrder

11.1.3.2.1 Description

This element defines the stacking order , relative to the *kml:AbstractViewGroup*, for overlapping *kml:AbstractOverlayGroup* elements. *kml:AbstractOverlayGroup* elements with higher kml:drawOrder values are drawn on top of overlays with lower kml:drawOrder values.

11.1.3.2.2 Content

Type:	xsd:int
Default Value:	0

11.1.3.3 kml:Icon

11.1.3.3.1 Description

Specifies the image associated with the *kml:AbstractOverlayGroup*. If no image is specified or located, a rectangle is drawn using the color and size defined by the ground or screen overlay.

See 13.1 kml:Link, kml:Icon (kml:LinkType).

11.1.3.4 kml:AbstractOverlaySimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.1.3.5 kml:AbstractOverlayObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.2 kml:GroundOverlay

11.2.1 Structure

```
<kml:GroundOverlay
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:drawOrder>...</kml:drawOrder> [0..1]
    <kml:Icon>...</kml:Icon> [0..1]
    <kml:AbstractOverlaySimpleExtensionGroup>...
     </kml:AbstractOverlaySimpleExtensionGroup> [0..*]
    <kml:AbstractOverlayObjectExtensionGroup>...
     </kml:AbstractOverlayObjectExtensionGroup> [0..*]
    <kml:altitude>...</kml:altitude> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:AbstractExtentGroup>...</kml:AbstractExtentGroup> [0..1]
    <kml:GroundOverlaySimpleExtensionGroup>...
     </kml:GroundOverlaySimpleExtensionGroup> [0..*]
    <kml:GroundOverlayObjectExtensionGroup>...
     </kml:GroundOverlayObjectExtensionGroup> [0..*]
</kml:GroundOverlay>
```

11.2.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractOverlayGroup*

Specifies how to display an image draped over the terrain. A kml:GroundOverlay element shall contain the kml:Icon and kml:LatLonBox child elements outside of an update context, that is when not a descendant of kml:Update.

11.2.3 Content

11.2.3.1 kml:altitude

11.2.3.1.1 Description

Specifies the distance above the terrain in meters. It shall be interpreted according to `kml:altitudeMode`. Only `kml:altitudeMode` **clampToGround** or **absolute** values shall be encoded for `kml:GroundOverlay`.

11.2.3.1.2 Content

Type:	xsd:double
Default Value:	0.0

11.2.3.2 kml:altitudeMode

See 9.20 kml:altitudeMode.

11.2.3.3 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

11.2.3.4 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.2.3.5 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.2.3.6 kml:AbstractExtentGroup

An abstract placeholder for `kml:LatLonBox` or `kml:LatLonQuad` in the context of `kml:GroundOverlay`.

See 9.17 kml:AbstractExtentGroup.

11.2.3.7 kml:GroundOverlaySimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.2.3.8 kml:GroundOverlayObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.2.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<GroundOverlay>
   <name>GroundOverlay.kml</name>
   <color>7fffffff</color>
   <drawOrder>1</drawOrder>
   <Icon>
      <href>http://www.google.com/intl/en/images/logo.gif</href>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>86400</refreshInterval>
      <viewBoundScale>0.75</viewBoundScale>
   </Icon>
   <LatLonBox>
      <north>37.83234</north>
      <south>37.832122</south>
      <east>-122.373033</east>
      <west>-122.373724</west>
      <rotation>45</rotation>
   </LatLonBox>
</GroundOverlay>
</kml>
```

11.3 kml:LatLonBox

11.3.1 Structure

```
<kml:LatLonBox
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentSimpleExtensionGroup>...
     </kml:AbstractExtentSimpleExtensionGroup> [0..*]
    <kml:AbstractExtentObjectExtensionGroup>...
     </kml:AbstractExtentObjectExtensionGroup> [0..*]
    <kml:north>...</kml:north> [0..1]
    <kml:south>...</kml:south> [0..1]
    <kml:east>...</kml:east> [0..1]
    <kml:west>...</kml:west> [0..1]
    <kml:AbstractLatLonBoxSimpleExtensionGroup>...
     </kml:AbstractLatLonBoxSimpleExtensionGroup> [0..*]
    <kml:AbstractLatLonBoxObjectExtensionGroup>...
     </kml:AbstractLatLonBoxObjectExtensionGroup> [0..*]
    <kml:rotation>...</kml:rotation> [0..1]
    <kml:LatLonBoxSimpleExtensionGroup>...
     </kml:LatLonBoxSimpleExtensionGroup> [0..*]
    <kml:LatLonBoxObjectExtensionGroup>...
     </kml:LatLonBoxObjectExtensionGroup> [0..*]
</kml:LatLonBox>
```

11.3.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Specifies the extent of a 2D bounding box. `kml:north`, `kml:south`, `kml:east`, and `kml:west` shall be specified.

11.3.3 Content

11.3.3.1 kml:rotation

11.3.3.1.1 Description

Specifies a rotation of the overlay about its center, in decimal degrees. Values can be ±180, with 0 being North. Rotations are specified in a counterclockwise direction.

11.3.3.1.2 Content

Type:	kml:angle180Type
Default Value:	0.0

11.3.3.2 kml:LatLonBoxSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.3.3.3 kml:LatLonBoxObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.3.4 Examples

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

```
<LatLonBox>
   <north>110</north>
   <south>0</south>
   <east>70</east>
   <west>-70</west>
   <rotation>0</rotation>
</LatLonBox>
```

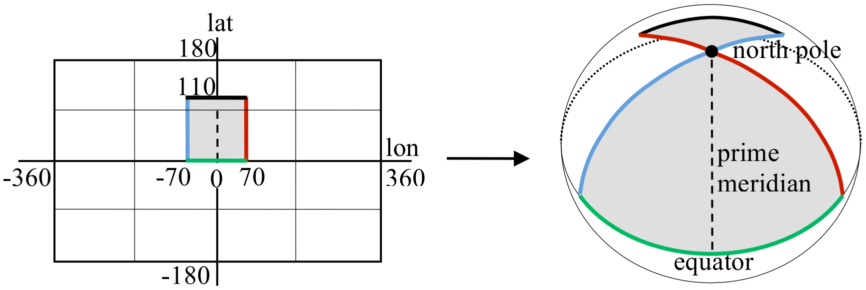


Figure 11: LatLonBox in (lon, lat) plane mapped to earth model

```
<LatLonBox>
   <north>90</north>
   <south>0</south>
   <east>360</east>
   <west>0</west>
   <rotation>0</rotation>
</LatLonBox>
```

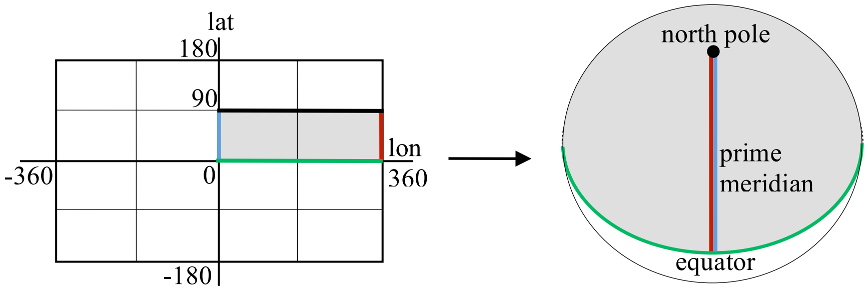


Figure 12: LatLonBox in (lon, lat) plane mapped to earth model

11.4 kml:LatLonQuad

11.4.1 Structure

```
<kml:LatLonQuad
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
   <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
   <kml:AbstractExtentSimpleExtensionGroup>...
    </kml:AbstractExtentSimpleExtensionGroup> [0..*]
   <kml:AbstractExtentObjectExtensionGroup>...
    </kml:AbstractExtentObjectExtensionGroup> [0..*]
   <kml:coordinates>...</kml:coordinates> [0..1]
   <kml:LatLonQuadSimpleExtensionGroup>...
    </kml:LatLonQuadSimpleExtensionGroup> [0..*]
  <kml:LatLonQuadObjectExtensionGroup>...
   </kml:LatLonQuadObjectExtensionGroup> [0..*]
</kml:LatLonQuad>
```

11.4.2 Description

The `kml:LatLonQuad` element supports general quadrilateral ground overlays (not necessarily rectangular). The coordinates of four corner points of the quadrilateral are specified in the `kml:coordinates` element. The coordinates shall be specified in counter-clockwise order with the first coordinate corresponding to the lower-left corner of the overlay. The shape described by these corners must be convex. If a third ordinate value is inserted into any tuple (representing altitude) it shall be ignored. Altitude is specified using the `kml:altitude` child element of `kml:GroundOverlay>` (a sibling element of `kml:LatLonQuad`). The permitted `kml:altitudeMode` values are: `absolute`, and `clampToGround` and the only valid `kml:seaFloorAltitudeMode` value is `clampToSeaFloor`. No other altitude mode values are permited for use in the context of `kml:LatLonQuad.`

11.4.3 Content

11.4.3.1 kml:coordinates

11.4.3.1.1 Description

Specifies four coordinate tuples, each consisting of double floating point values for longitude and latitude. A singlespace separates the tuples and commas separate the ordinates.

11.4.3.1.2 Content

Type:	kml:coordinatesType
Default Value:	none

11.4.3.2 kml:LatLonQuadSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.4.3.3 kml:LatLonQuadObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.4.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <GroundOverlay>
    <name>LatLonQuad Example</name>
    <Icon>
      <href>http://developers.google.com/kml/documentation/images/rectangle.gif
       </href>
      <viewBoundScale>0.75</viewBoundScale>
    </Icon>
    <LatLonQuad>
      <coordinates>81.601884,44.160723 83.529902,43.665148 82.947737,44.248831
81.509322,44.321015</coordinates>
    </LatLonQuad>
  </GroundOverlay>
</kml>
```

11.5 kml:PhotoOverlay

11.5.1 Structure

```
<kml:PhotoOverlay
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:drawOrder>...</kml:drawOrder> [0..1]
    <kml:Icon>...</kml:Icon> [0..1]
    <kml:AbstractOverlaySimpleExtensionGroup>...
     </kml:AbstractOverlaySimpleExtensionGroup> [0..*]
    <kml:AbstractOverlayObjectExtensionGroup>...
     </kml:AbstractOverlayObjectExtensionGroup> [0..*]
    <kml:rotation>...</kml:rotation> [0..1]
    <kml:ViewVolume>...</kml:ViewVolume> [0..1]
    <kml:ImagePyramid>...</kml:ImagePyramid> [0..1]
    <kml:Point>...</kml:Point> [0..1]
    <kml:abstractShape>...</kml:abstractShape> [0..1]
    <kml:PhotoOverlaySimpleExtensionGroup>...
     </kml:PhotoOverlaySimpleExtensionGroup> [0..*]
    <kml:PhotoOverlayObjectExtensionGroup>...
     </kml:PhotoOverlayObjectExtensionGroup> [0..*]
</kml:PhotoOverlay>
```

11.5.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractOverlayGroup*

The kml:PhotoOverlay element is used to position a photograph relative to the camera viewpoint and also to define field-of-view parameters. The kml:PhotoOverlay may be rendered on one of three shapes (as determined by the value of kml:shape): a 2D planar rectangle, a cylinder (for a panoramic photo), or a sphere (for a spherical panorama). The kml:PhotoOverlay is projected onto the kml:shape as shown in the case of a cylinder in Figure 10.

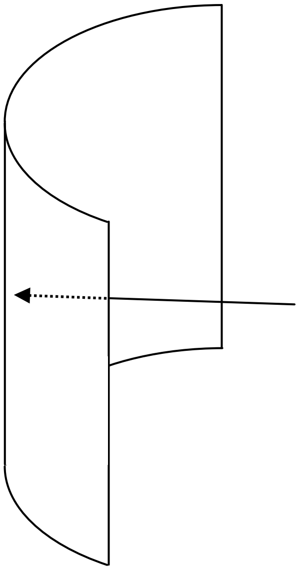


Figure 13: A Cylindrical kml:shape Positioned Relative to View Vector

The the photo overlay faces toward the viewpoint and its center is placed at the head of the view vector. The view vector is defined as the vector based at the viewpoint and in the direction specified by the corresponding kml:AbstractView element. The length of the view vector is determined by the value of the kml:near element. The photo overlay is positioned such that the view vector points toward the photo and is orthogonal to the center of the image (see Figure 14).

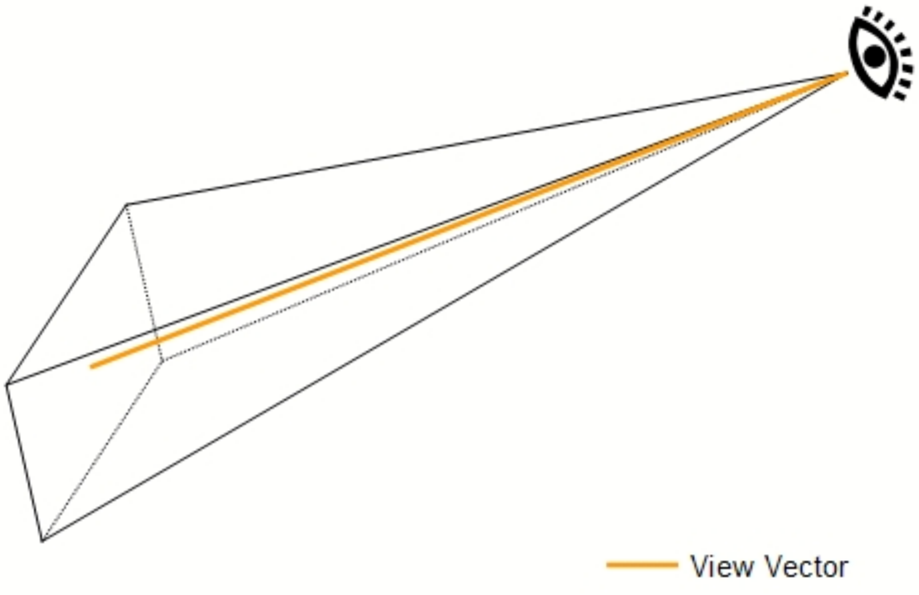


Figure 14: kml:PhotoOverlay View Vector

The URL for the `kml:PhotoOverlay` image is specified in the `kml:Icon` tag, which is inherited from *`kml:AbstractOverlayGroup`*. The `kml:Icon` tag must contain a `kml:href` element that specifies the image resource to use for the `kml:PhotoOverlay`.

A `kml:PhotoOverlay` element shall contain the `kml:Icon`, `kml:ViewVolume`, `kml:Point`, and `kml:Camera` child elements outside of an update context, that is when not a descendant of `kml:Update`.

11.5.2.1 Handling large images

For large images an image pyramid (`kml:ImagePyramid`) may be used to specify a hierarchical set of images, each of which is an increasingly lower resolution version of the original image. Each image in the pyramid is subdivided into tiles, so that only the portions in view need to be loaded. An earth browser calculates the current viewpoint and load the tiles that are appropriate to the user's distance from the image. As the viewpoint moves closer to the `kml:PhotoOverlay`, the earth browser loads the higher resolution tiles. Since all the pixels in the original image can't be viewed on the screen at once, this preprocessing allows an earth browser to achieve maximum performance because it loads only the portions of the image that are in view, and only the pixel details that can be discerned by the user at the current viewpoint.

When a `kml:ImagePyramid` is present, the `kml:href` specification in the `kml:Icon` element shall include parameterization to specify the *level*, *x*, and *y* values of the tiles to fetch, where:

- *x* = row position in the grid
- *y* = column position in the grid

- *level* = level in the image pyramid, with 0 being the highest level

For example, the URL for the image might be specified as follows:

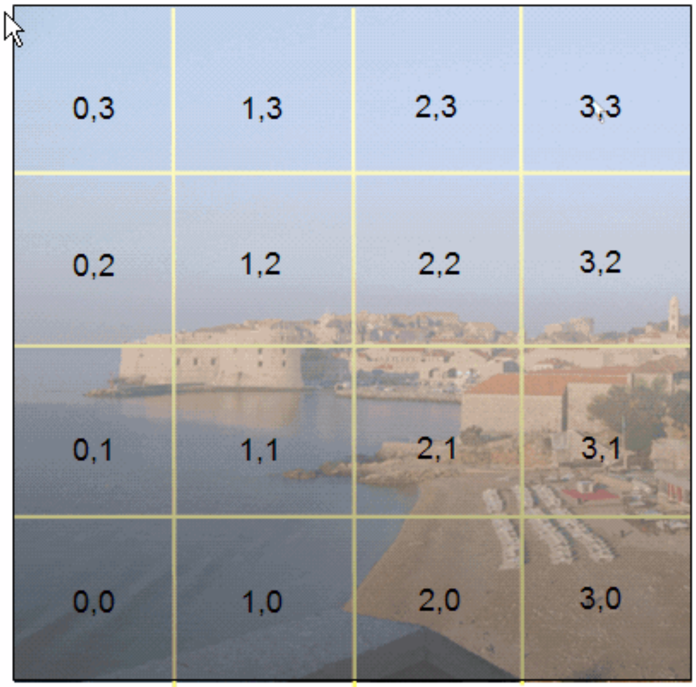
```
http://server.company.com/bigphoto/$[level]/row_$[x]_column_[$y].jpg
```

To request the tile in row 2, column 1, at level 3, Google Earth would fetch the following URL:

```
http://server.company.com/bigphoto/3/row_2_column_1.jpg
```

By default, the origin (0,0) is at the lower left of the grid. If an image has an origin in the upper left, the `kml:ImagePyramid` shall specify **topLeft** for the `kml:gridOrigin`.

The following figure illustrates numbering of tiles at level 2 of a 10-megapixel image:



Level 2 *(numbering the tiles)*

See also 13.1 kml:Link, kml:Icon (kml:LinkType), 11.8.3 Creating an Image Pyramid.

11.5.3 Content

11.5.3.1 kml:rotation

See 11.3.3.1 kml:rotation.

11.5.3.2 kml:ViewVolume

11.5.3.2.1 Description

Defines how much of the current scene is visible.

See 11.7 kml:ViewVolume.

11.5.3.3 kml:ImagePyramid

11.5.3.3.1 Description

See 11.8 kml:ImagePyramid.

11.5.3.4 kml:Point

Specifies the location of an icon associated with the `kml:PhotoOverlay`. The `kml:Point` is styled using associated or default styles.

See 10.3 kml:Point.

11.5.3.5 kml:abstractShape

11.5.3.5.1 Description

An abstract placeholder for `kml:shape` and an extension point for shape enumerations. See also 11.6 kml:shape and 6.7.2.1 Simple Element Substitution.

11.5.3.5.2 Content

Type:	kml:enumBaseType
Default Value:	none

11.5.3.6 kml:PhotoOverlaySimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.5.3.7 kml:PhotoOverlayObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.5.4 Example

```
<PhotoOverlay>
  <!-- Feature elements -->
  <name>A simple non-pyramidal photo</name>
  <description>High above the ocean</description>
  <!-- AbstractOverlayGroup elements -->
  <Icon>
  <!-- A simple normal jpeg image -->
  <href>small-photo.jpg</href>
  </Icon>
  <!-- PhotoOverlay elements -->
  <!-- default: <shape> -->
  <ViewVolume>
    <near>1000</near>
    <leftFov>-60</leftFov>
    <rightFov>60</rightFov>
    <bottomFov>-45</bottomFov>
    <topFov>45</topFov>
  </ViewVolume>
  <Point>
    <coordinates>1,1</coordinates>
  </Point>
  <!-- if no ImagePyramid only level 0 is shown,
       fine for a non-pyramidal image -->
</PhotoOverlay>
```

11.6 kml:shape

11.6.1 Structure

```
<kml:shape>...</kml:shape>
```

11.6.2 Description

Specifies a shape using an enumerated value.

11.6.3 Content

Type:	kml:shapeEnumType
Default Value:	**rectangle**

11.7 kml:ViewVolume

11.7.1 Structure

```
<kml:ViewVolume
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:leftFov>...</kml:leftFov> [0..1]
    <kml:rightFov>...</kml:rightFov> [0..1]
    <kml:bottomFov>...</kml:bottomFov> [0..1]
    <kml:topFov>...</kml:topFov> [0..1]
    <kml:near>...</kml:near> [0..1]
    <kml:ViewVolumeSimpleExtensionGroup>...
     </kml:ViewVolumeSimpleExtensionGroup> [0..*]
    <kml:ViewVolumeObjectExtensionGroup>...
     </kml:ViewVolumeObjectExtensionGroup> [0..*]
</kml:ViewVolume>
```

11.7.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Defines how much of the current scene in a kml:PhotoOverlay is visible. Specifying the field of view is analogous to specifying the lens opening in a physical camera. A small field of view, like a telephoto lens, focuses on a small part of the scene. A large field of view, like a wide-angle lens, focuses on a large part of the scene.

A kml:ViewVolume element shall contain the kml:leftFov, kml:rightFov, kml:bottomFov, kml:topFov, and kml:near child elements outside of an update context, that is when not a descendant of kml:Update.

11.7.3 Content

11.7.3.1 kml:leftFov

11.7.3.1.1 Description

Angle, in decimal degrees, from the left side of the view volume to the camera's view vector. A negative value of the angle corresponds to a field of view that is ‘left’ of the view vector.

11.7.3.1.2 Content

Type:	kml:angle180Type
Default Value:	0.0

11.7.3.2 kml:rightFov

11.7.3.2.1 Description

Angle, in decimal degrees, from the camera's view vector to the right side of the view volume. A positive value of the angle corresponds to a field of view that is ‘right’ of the view vector.

11.7.3.2.2 Content

Type:	kml:angle180Type
Default Value:	0.0

11.7.3.3 kml:bottomFov

11.7.3.3.1 Description

Angle, in decimal degrees, from the the bottom side of the view volume to camera's view vector.

11.7.3.3.2 Content

Type:	kml:angle90Type
Default Value:	0.0

11.7.3.4 kml:topFov

11.7.3.4.1 Description

Angle, in decimal degrees, from the camera's view vector to the top side of the view volume.

11.7.3.4.2 Content

Type:	kml:angle90Type
Default Value:	0.0

11.7.3.5 kml:near

11.7.3.5.1 Description

Length in meters of the view vector, which starts from the camera viewpoint and ends at the `kml:PhotoOverlay` shape. The value shall be positive.

11.7.3.5.2 Content

Type:	xsd:double
Default Value:	0.0

11.7.3.6 kml:ViewVolumeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.7.3.7 kml:ViewVolumeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.8 kml:ImagePyramid

11.8.1 Structure

```
<kml:ImagePyramid
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:tileSize>...</kml:tileSize> [0..1]
    <kml:maxWidth>...</kml:maxWidth> [0..1]
    <kml:maxHeight>...</kml:maxHeight> [0..1]
    <kml:abstractGridOrigin>...</kml:abstractGridOrigin> [0..1]
    <kml:ImagePyramidSimpleExtensionGroup>...
     </kml:ImagePyramidSimpleExtensionGroup> [0..*]
    <kml:ImagePyramidObjectExtensionGroup>...
     </kml:ImagePyramidObjectExtensionGroup> [0..*]
</kml:ImagePyramid>
```

11.8.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Specifies a hierarchical set of images, each of which is an increasingly lower resolution (towards the top of the pyramid). Each image in the pyramid is subdivided into tiles so only the portions in view are loaded.

The pixel size of the original image is specified in the `kml:maxWidth` and `kml:maxHeight` elements. The width and height can be any size and do not need to be a power of 2. You can fill out the remaining pixels with blank pixels, as described in.

Tiles must be square, and the `kml:tileSize` must be a power of 2. A tile size of 256 (the default) or 512 is recommended.

A `kml:ImagePyramid` element should have the `kml:maxWidth` and `kml:maxHeight` child elements present, and, if supplied, the `kml:tileSize` should be a power of 2.

See also 11.5.2.1 Handling large images.

11.8.3 Creating an Image Pyramid

An image pyramid should be constructed as follows, assuming the image pixel measurement is a power of 2.

1. Starting with the original, full-size image, divide it into tile-sized pieces—for example, into blocks of 256 * 256 pixels each.
2. Shrink the image by a factor of 2.
3. Divide this new image into tile-sized squares.
4. Repeat steps 2 and 3 until the resulting image fits inside the tile size (for example, 256 * 256 pixels).

If the image pixel measurement is not a power of 2, transparent fill pixels shall be added to make the tile square. Place the image so that the (0,0) tile is at the origin. For example, if the origin is at the lower left, the image should be located in the lower left of the tile grid. The row and columns that might need fill would then be at the right and top of the image. For best filtering, replicate the last row (or column) at the edge of the image. Then add fill (for example, black) to the remaining pixels in the tiles of the row (or column).

For example, consider an image whose dimensions are 3600 * 2700 pixels (roughly 10 megapixels). An image pyramid for this image should be created as follows:

1. Using a tile size of 256 pixels, you can subdivide the original image into a grid of 16 * 16 pixels. (This image ends up as level 4 in the final pyramid.)
2. Fill in the pixels to "square up" the partially filled tiles in the last column (to the right) and the last row (at the top, assuming `kml:gridOrigin` is **lowerLeft**).
3. Scale down the image by a factor of 2.
4. Subdivide this image into 256-pixel tiles. The image at this level consists of a grid of 8 * 8 tiles (level 3).
5. Scale the level 3 image down by a factor of 2.
6. Subdivide into tiles. The image at this level consists of a grid of 4 * 4 tiles (level 2).
7. Scale the level 2 image down by a factor of 2.
8. Subdivide into tiles. The image at this level consists of a grid of 2 * 2 tiles (level 1).
9. Scale the level 1 image down by a factor of 2.
10. The resulting image is 256 * 256 pixels, so this is the last level of the image pyramid (level 0).

The image pyramid for a 4096 * 4096 image has 5 levels, as shown in Table 4.

Table 4: Example of Image Pyramid Levels

Level	**Number of Tiles**	**Size of Image (pixels)**
0	1	256 * 256

1	4 (2 * 2 grid)	512 * 512
2	16 (4 * 4 grid)	1024 * 1024
3	64 (8 * 8 grid)	2048 * 2048
4	256 (16 * 16 grid)	4096 * 4096

Level *n* thus has 2^n tiles in each direction.

11.8.4 Transparency

If an image is fully opaque then the image should be encoded in JPEG format. If part of the image is opaque and part is transparent both PNG and JPEG tiles may be specified, with PNG used for tiles that have transparency values. If both formats are used omit the file extension from the `kml:href` specification of the image file and include the file extension in the filename for each tile.

11.8.5 Content

11.8.5.1 kml:tileSize

11.8.5.1.1 Description

Size of the tiles, in pixels. Tiles must be square, and `kml:tileSize` must be a power of 2. A tile size of 256 (the default) or 512 is recommended. The original image is divided into tiles of this size, at varying resolutions.

11.8.5.1.2 Content

Type:	xsd:int
Default Value:	256

11.8.5.2 kml:maxWidth

11.8.5.2.1 Description

Width in pixels of the original image.

11.8.5.2.2 Content

Type:	xsd:int
Default Value:	0

11.8.5.3 kml:maxHeight

11.8.5.3.1 Description

Height in pixels of the original image.

11.8.5.3.2 Content

Type:	xsd:int
Default Value:	0

11.8.5.4 kml:abstractGridOrigin

11.8.5.4.1 Description

An abstract placeholder for `kml:gridOrigin` and an extension point for grid origin enumerations. See also 11.9 kml:gridOrigin and 6.7.2.1 Simple Element Substitution.

11.8.5.4.2 Content

Type:	kml:enumBaseType
Default Value:	none

11.8.5.5 kml:ImagePyramidSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

11.8.5.6 kml:ImagePyramidObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

11.9 kml:gridOrigin

11.9.1 Structure

```
<kml:gridOrigin>...</kml:gridOrigin>
```

11.9.2 Description

Specifies where to begin numbering the tiles in each layer of the pyramid. A value of `lowerLeft` specifies that row 1, column 1 of each layer is in the bottom left corner of the grid.

11.9.3 Content

Type:

kml:gridOriginEnumType
Default Value: **lowerLeft**

11.10 kml:ScreenOverlay

11.10.1 Structure

```
<kml:ScreenOverlay
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:name>...</kml:name> [0..1]
    <kml:visibility>...</kml:visibility> [0..1]
    <kml:balloonVisibility>...</kml:balloonVisibility> [0..1]
    <kml:open>...</kml:open> [0..1]
    <atom:author>...</atom:author> [0..1]
    <atom:link>...</atom:link> [0..1]
    <kml:address>...</kml:address> [0..1]
    <xal:AddressDetails>...</xal:AddressDetails> [0..1]
    <kml:phoneNumber>...</kml:phoneNumber> [0..1]
    <kml:snippet>...</kml:snippet> [0..1]
    <kml:description>...</kml:description> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..*]
    <kml:Region>...</kml:Region> [0..1]
    <kml:AbstractExtendedDataGroup>...</kml: AbstractExtendedDataGroup> [0..1]
    <kml:AbstractFeatureSimpleExtensionGroup>...
     </kml:AbstractFeatureSimpleExtensionGroup> [0..*]
    <kml:AbstractFeatureObjectExtensionGroup>...
     </kml:AbstractFeatureObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:drawOrder>...</kml:drawOrder> [0..1]
    <kml:Icon>...</kml:Icon> [0..1]
    <kml:AbstractOverlaySimpleExtensionGroup>...
     </kml:AbstractOverlaySimpleExtensionGroup> [0..*]
    <kml:AbstractOverlayObjectExtensionGroup>...
     </kml:AbstractOverlayObjectExtensionGroup> [0..*]
    <kml:overlayXY>...</kml:overlayXY> [0..1]
    <kml:screenXY>...</kml:screenXY> [0..1]
    <kml:rotationXY>...</kml:rotationXY> [0..1]
    <kml:size>...</kml:size> [0..1]
    <kml:rotation>...</kml:rotation> [0..1]
    <kml:ScreenOverlaySimpleExtensionGroup>...
     </kml:ScreenOverlaySimpleExtensionGroup> [0..*]
    <kml:ScreenOverlayObjectExtensionGroup>...
     </kml:ScreenOverlayObjectExtensionGroup> [0..*]
</kml:ScreenOverlay>
```

11.10.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractOverlayGroup*

Specifies an image overlay to be displayed fixed to the screen. The image position is determined by mapping a point relative to the image (specified by `kml:overlayXY`) to a point relative to the screen (specified by `kml:screenXY`). The image may be rotated by `kml:rotation` degrees about a point relative to the screen (specified by `kml:rotationXY`). The image sizing is determined using the `kml:size` element.

A `kml:ScreenOverlay` element should contain the `kml:screenXY` element outside of an update context, that is when not a descendant of `kml:Update`.

11.10.3 Content

11.10.3.1 kml:overlayXY

11.10.3.1.1 Description

Specifies a point on (or outside of) the image that is mapped to the screen coordinate (`kml:screenXY`). It requires *x* and *y* values, and the units for those values. The origin of the coordinate system is the lower left corner of the icon.

11.10.3.1.2 Content

Type:	kml:vec2Type
Default Value:	See kml:vec2Type

11.10.3.2 kml:screenXY

11.10.3.2.1 Description

Specifies a point relative to the screen origin that the image is mapped to. It requires *x* and *y* values, and the units for those values. The origin of the coordinate system is the lower left corner of the screen.

11.10.3.2.2 Content

Type:	kml:vec2Type
Default Value:	See kml:vec2Type

11.10.3.2.3 Example

11.10.3.3 kml:rotationXY

11.10.3.3.1 Description

Point relative to the screen about which the screen overlay is rotated. The origin of the coordinate system is in the lower left corner of the screen.

11.10.3.3.2 Content

Type:	kml:vec2Type
Default Value:	See kml:vec2Type

11.10.3.4 kml:size

11.10.3.4.1 Description

Specifies the size of the image for a `kml:ScreenOverlay`. A value of -1 indicates to use the native dimension; a value of 0 indicates to maintain the aspect ratio; a value of *n* sets the value of the dimension.

11.10.3.4.2 Content

Type:	kml:vec2Type
Default Value:	See kml:vec2Type

11.10.3.4.3 Example

To force the image to retain its original x and y dimensions, set the values to −1:

```
<size x="-1" y="-1" xunits="fraction" yunits="fraction"/>
```

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

```
<size x="-1" 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"/>
```

11.10.3.5 kml:rotation

11.10.3.5.1 Description

Indicates the angle of rotation, in decimal degrees, of the parent object. A value of 0 means no rotation. The value is an angle in decimal degrees counterclockwise starting from north. Use ±180 to indicate the rotation of the parent object from 0. The center of the `kml:rotation`, if not (.5,.5), is specified in `kml:rotationXY`.

11.10.3.5.2 Content

Type:	kml:angle180Type
Default Value:	0.0

11.10.4 Examples

To center an image on the screen:

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

To place an image on the top left of the screen:

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

To place an 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>
```

The following example places an image at the exact center of the screen, using the original width, height, and aspect ratio of the image.

```
<ScreenOverlay id="khScreenOverlay756">
  <name>Simple crosshairs</name>
  <description>This screen overlay uses fractional positioning
   to put the image in the exact center of the screen</description>
  <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>
```

12. Styles

12.1 kml:AbstractStyleSelectorGroup

12.1.1 Structure

```
<kml:AbstractStyleSelectorGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorSimpleExtensionGroup>...
     </kml:AbstractStyleSelectorSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorObjectExtensionGroup>...
     </kml:AbstractStyleSelectorObjectExtensionGroup> [0..*]
</kml:AbstractStyleSelectorGroup>
```

12.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:Style
- kml:StyleMap

12.1.3 Content

12.1.3.1 kml:AbstractStyleSelectorSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.1.3.2 kml:AbstractStyleSelectorObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.2 kml:Style

12.2.1 Structure

```
<kml:Style
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorSimpleExtensionGroup>...
     </kml:AbstractStyleSelectorSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorObjectExtensionGroup>...
     </kml:AbstractStyleSelectorObjectExtensionGroup> [0..*]
    <kml:IconStyle>...</kml:IconStyle> [0..1]
    <kml:LabelStyle>...</kml:LabelStyle> [0..1]
    <kml:LineStyle>...</kml:LineStyle> [0..1]
    <kml:PolyStyle>...</kml:PolyStyle> [0..1]
    <kml:BalloonStyle>...</kml:BalloonStyle> [0..1]
    <kml:ListStyle>...</kml:ListStyle> [0..1]
    <kml:StyleSimpleExtensionGroup>...</kml:StyleSimpleExtensionGroup> [0..*]
    <kml:StyleObjectExtensionGroup>...</kml:StyleObjectExtensionGroup> [0..*]
</kml:Style>
```

12.2.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractStyleSelectorGroup*

Specifies a container of zero or more kml:AbstractColorStyleGroup elements that can referenced from a kml:StyleMap or *kml:AbstractFeatureGroup*. Styles affect how a *kml:AbstractGeometryGroup* is presented in the geographic view and how a *kml:AbstractFeatureGroup* appears in the list view.

kml:Scale should contain at least one child element outside of an update context, that is when not a descendant of kml:Update. It is advised that kml:x, kml:y, and kml:z all be specified.

12.2.3 Content

12.2.3.1 kml:IconStyle

See 12.12 kml:IconStyle.

12.2.3.2 kml:LabelStyle

See 12.14 kml:LabelStyle.

12.2.3.3 kml:LineStyle

See 12.15 kml:LineStyle.

12.2.3.4 kml:PolyStyle

See 12.16 kml:PolyStyle.

12.2.3.5 kml:BalloonStyle

See 12.7 kml:BalloonStyle.

12.2.3.6 kml:ListStyle

See 12.17 kml:ListStyle.

12.2.3.7 kml:StyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.2.3.8 kml:StyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.2.4 Example

```
<Document>
  <!-- Begin Style Definitions -->
  <Style id="myDefaultStyles">
    <IconStyle>
      <color>a1ff00ff</color>
      <scale>1.399999976158142</scale>
      <Icon>
        <href>http://myserver.com/icon.jpg</href>
      </Icon>
    </IconStyle>
    <LabelStyle>
      <color>7fffaaff</color>
      <scale>1.5</scale>
    </LabelStyle>
    <LineStyle>
      <color>ff0000ff</color>
      <width>15</width>
    </LineStyle>
    <PolyStyle>
      <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>
```

12.3 kml:StyleMap

12.3.1 Structure

```
<kml:StyleMap
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorSimpleExtensionGroup>...
     </kml:AbstractStyleSelectorSimpleExtensionGroup> [0..*]
    <kml:AbstractStyleSelectorObjectExtensionGroup>...
     </kml:AbstractStyleSelectorObjectExtensionGroup> [0..*]
    <kml:Pair>...</kml:Pair> [0..*]
    <kml:StyleMapSimpleExtensionGroup>...</kml:StyleMapSimpleExtensionGroup>
[0..*]
    <kml:StyleMapObjectExtensionGroup>...</kml:StyleMapObjectExtensionGroup>
[0..*]
</kml:StyleMap>
```

12.3.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractStyleSelectorGroup*

Specifies a mapping between two kml:Styles using a key/value pair that maps a mode to the predefined kml:styleUrl. A kml:StyleMap may be used to provide separate normal and highlighted styles for a kml:Placemark.

kml:StyleMap should have two kml:Pair elements, one with a kml:key value of **normal** and the other with a value of **highlight,** outside of an update context, that is when not a descendant of kml:Update.

12.3.3 Content

12.3.3.1 kml:Pair

See 12.4 kml:Pair.

12.3.3.2 kml:StyleMapSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.3.3.3 kml:StyleMapObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.3.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>StyleMap.kml</name>
  <open>1</open>
  <Style id="normalState">
    <IconStyle>
      <scale>1.0</scale>
      <Icon>
        <href>http://maps.google.com/mapfiles/kml/pal3/icon55.png</href>
      </Icon>
    </IconStyle>
    <LabelStyle>
      <scale>1.0</scale>
    </LabelStyle>
  </Style>
  <Style id="highlightState">
    <IconStyle>
      <Icon>
        <href>http://maps.google.com/mapfiles/kml/pal3/icon60.png</href>
      </Icon>
      <scale>1.1</scale>
    </IconStyle>
    <LabelStyle>
      <scale>1.1</scale>
      <color>ff0000c0</color>
    </LabelStyle>
  </Style>
  <StyleMap id="styleMapExample">
    <Pair>
      <key>normal</key>
      <styleUrl>#normalState</styleUrl>
    </Pair>
    <Pair>
      <key>highlight</key>
      <styleUrl>#highlightState</styleUrl>
    </Pair>
  </StyleMap>
  <Placemark>
    <name>StyleMap example</name>
    <styleUrl>#styleMapExample</styleUrl>
    <Point>
      <coordinates>-122.368987,37.817634,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

12.4 kml:Pair

12.4.1 Structure

```
<kml:Pair
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:abstractKey>...</kml:abstractKey> [0..1]
    <kml:styleUrl>...</kml:styleUrl> [0..1]
    <kml:AbstractStyleSelectorGroup>...</kml:AbstractStyleSelectorGroup> [0..1]
    <kml:PairSimpleExtensionGroup>...</kml:PairSimpleExtensionGroup> [0..*]
    <kml:PairObjectExtensionGroup>...</kml:PairObjectExtensionGroup> [0..*]
</kml:Pair>
```

12.4.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Defines a key/value pair that maps a mode (normal or highlight) to the predefined kml:styleUrl and/or a kml:AbstractStyleSelectorGroup. If both kml:styleUrl and kml:AbstractStyleSelectorGroup exist then their styles shall be merged.

A kml:Pair element shall contain kml:key and either kml:styleUrl or kml:AbstractStyleSelectorGroup child elements outside of an update context, that is when not a descendant of kml:Update.

12.4.3 Content

12.4.3.1 kml:abstractKey

12.4.3.1.1 Description

An abstract placeholder for kml:key and an extension point for key enumerations. See also 12.5 kml:key and 6.7.2.1 Simple Element Substitution.

12.4.3.1.2 Content

Type:	kml:enumBaseType
Default Value:	none

12.4.3.2 kml:styleURL

See 9.1.3.14 kml:styleURL.

12.4.3.3 kml:AbstractStyleSelectorGroup

See 12.1 kml:AbstractStyleSelectorGroup, 12.2 kml:Style and 12.3 kml:StyleMap.

12.4.3.4 kml:PairSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.4.3.5 kml:PairObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.4.4 Example

```
<Pair>
  <key>normal</key>
  <styleUrl>http://myserver.com/population.xml#example_style_off</styleUrl>
</Pair>
```

12.5 kml:key

12.5.1 Structure

```
<kml:key>...</kml:key>
```

12.5.2 Description

Identifies a key whose value is either **normal** or **highlight**.

12.5.3 Content

Type:	kml:styleStateEnumType
Default Value:	**normal**

12.6 kml:AbstractSubStyleGroup

12.6.1 Structure

```
<kml:AbstractSubStyleGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
</kml:AbstractSubStyleGroup>
```

12.6.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- *kml:AbstractColorStyleGroup*
- kml:BalloonStyle
- kml:IconStyle
- kml:ListStyle
- kml:LabelStyle
- kml:LineStyle
- kml:PolyStyle

12.6.3 Content

12.6.3.1 kml:AbstractSubStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.6.3.2 kml:AbstractSubStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.7 kml:BalloonStyle

12.7.1 Structure

```
<kml:BalloonStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:AbstractBgColorGroup>...</kml:AbstractBgColorGroup> [0..1]
    <kml:textColor>...</kml:textColor> [0..1]
    <kml:text>...</kml:text> [0..1]
    <kml:abstractDisplayMode>...</kml:abstractDisplayMode> [0..1]
    <kml:BalloonStyleSimpleExtensionGroup>...
     </kml:BalloonStyleSimpleExtensionGroup> [0..*]
    <kml:BalloonStyleObjectExtensionGroup>...
     </kml:BalloonStyleObjectExtensionGroup> [0..*]
</kml:BalloonStyle>
```

12.7.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractSubStyleGroup*

Specifies how the description balloon for a *kml:AbstractFeatureGroup* is drawn.

kml:BalloonStyle should contain at least one child element outside of an update context, that is when not a descendant of kml:Update.

12.7.3 Content

12.7.3.1 kml:AbstractBgColorGroup

12.7.3.1.1 Description

An abstract placeholder for kml:bgColor in the context of kml:BalloonStyle.

See 12.8 kml:bgColor and 6.7.2.1 Simple Element Substitution.

12.7.3.1.2 Content

Type:	kml:colorType
Default Value:	none

12.7.3.2 kml:textColor

12.7.3.2.1 Description

Specifies the foreground color of the text.

12.7.3.2.2 Content

Type:	kml:colorType
Default Value:	**ffffffff**

12.7.3.3 kml:text

12.7.3.3.1 Description

Specifies the text displayed in the balloon.

The text may include HTML content that is encoded as well-formed XML using HTML entity references (e.g. '&lt;', '&gt;', etc) or by enclosing the HTML within a CDATA section. Note that KML software implementations are unlikely to support all of HTML, CSS, and JavaScript.

`kml:text` shall support entity substitution as defined in 6.5 Entity Replacement.

12.7.3.3.2 Content

Type:	xsd:string
Default Value:	none

12.7.3.4 kml:abstractDisplayMode

12.7.3.4.1 Description

An abstract placeholder for `kml:displayMode` and an extension point for display mode enumerations. See also 12.9 kml:displayMode and 6.7.2.1 Simple Element Substitution.

12.7.3.4.2 Content

Type:	kml:enumBaseType
Default Value:	none

12.7.3.5 kml:BalloonStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.7.3.6 kml:BalloonStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.7.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>BalloonStyle.kml</name>
  <open>1</open>
  <Style id="exampleBalloonStyle">
    <BalloonStyle>
      <!-- a background color for the balloon -->
      <bgColor>ffffffbb</bgColor>
      <!-- styling of the balloon text -->
      <text><![CDATA[
      <b><font color="#CC0000" size="+3">$[name]</font></b>
      <br/><br/>
      <font face="Courier">$[description]</font>
      <br/><br/>
      Extra text that will appear in the description balloon
      <br/><br/>
      <!-- insert the to/from hyperlinks -->
      $[geDirections]
      ]]></text>
    </BalloonStyle>
  </Style>
  <Placemark>
    <name>BalloonStyle</name>
    <description>An example of BalloonStyle</description>
    <styleUrl>#exampleBalloonStyle</styleUrl>
    <Point>
      <coordinates>-122.370533,37.823842,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

12.8 kml:bgColor

12.8.1 Description

Specifies the background color of the graphic element.

12.8.2 Content

Type:	kml:colorType
Default Value:	**ffffffff**

12.9 kml:displayMode

12.9.1 Structure

```
<kml:displayMode...</kml:displayMode>
```

12.9.2 Description

Controls whether the balloon is displayed or hidden. If `kml:displayMode` is **default**, the balloon shall be displayed. If `kml:displayMode` is **hide**, the balloon shall not be displayed.

12.9.3 Content

Type:	kml:displayModeEnumType
Default Value:	**default**

12.10 kml:AbstractColorStyleGroup

12.10.1 Structure

```
<kml:AbstractColorStyleGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:abstractColorMode>...</kml:abstractColorMode> [0..1]
    <kml:AbstractColorStyleSimpleExtensionGroup>...
     </kml:AbstractColorStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractColorStyleObjectExtensionGroup>...
     </kml:AbstractColorStyleObjectExtensionGroup> [0..*]
</kml:AbstractColorStyleGroup>
```

12.10.2 Description

This abstract element can be used wherever the following element is referenced:

- *`kml:AbstractSubStyleGroup`*

The following elements can be used wherever this abstract element is referenced:

- `kml:IconStyle`
- `kml:LineStyle`
- `kml:LabelStyle`
- `kml:PolyStyle`

Provides elements for specifying the color and color mode of style types that derive from it.

12.10.3 Content

12.10.3.1 kml:color

12.10.3.1.1 Description

Specifies the color of the graphic element.

12.10.3.1.2 Content

Type:	kml:colorType
Default Value:	**ffffffff**

12.10.3.2 kml:abstractColorMode

12.10.3.2.1 Description

An abstract placeholder for `kml:colorMode` and an extension point for color mode enumerations. See also 12.11 kml:colorMode and 6.7.2.1 Simple Element Substitution.

12.10.3.2.2 Content

Type:	kml:enumBaseType
Default Value:	none

12.10.3.3 kml:AbstractColorStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.10.3.4 kml:AbstractColorStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.11 kml:colorMode

12.11.1 Structure

```
<kml:colorMode...</kml:colorMode>
```

12.11.2 Description

Specifies the color mode of the graphic element.

12.11.3 Content

Type:	kml:colorModeEnumType
Default Value:	**normal**

12.12 kml:IconStyle

12.12.1 Structure

```
<kml:IconStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:abstractColorMode>...</kml:abstractColorMode> [0..1]
    <kml:AbstractColorStyleSimpleExtensionGroup>...
     </kml:AbstractColorStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractColorStyleObjectExtensionGroup>...
     </kml:AbstractColorStyleObjectExtensionGroup> [0..*]
    <kml:scale>...</kml:scale> [0..1]
    <kml:heading>...</kml:heading> [0..1]
    <kml:Icon> kml:BasicLinkType </kml:Icon> [0..1]
    <kml:hotSpot>...</kml:hotSpot> [0..1]
    <kml:IconStyleSimpleExtensionGroup>...
     </kml:IconStyleSimpleExtensionGroup> [0..*]
    <kml:IconStyleObjectExtensionGroup>...
     </kml:IconStyleObjectExtensionGroup> [0..*]
</kml:IconStyle>
```

12.12.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractColorStyleGroup*

Specifies how icons for kml:Placemarks and kml:PhotoOverlay with a kml:Point geometry are drawn in an earth browser's list and geographic views. The color specified in the kml:color element of kml:IconStyle is blended with the color of the icon.

kml:IconStyle should contain at least one child element outside of an update context, that is when not a descendant of kml:Update.

12.12.3 Content

12.12.3.1 kml:scale

12.12.3.1.1 Description

Specifies a scale factor that shall be applied to the graphic element.

12.12.3.1.2 Content

Type:	xsd:double
Default Value:	1.0

12.12.3.2 kml:heading

12.12.3.2.1 Description

Direction (North, South, East, West), in decimal degrees. Values range from 0 (North) to 360 degrees.

12.12.3.2.2 Content

Type:	kml:angle360Type
Default Value:	0.0

12.12.3.3 kml:Icon

See 12.13 kml:Icon (kml:BasicLinkType).

12.12.3.4 kml:hotSpot

12.12.3.4.1 Description

Specifies the position of the reference point on the icon that is anchored to the `kml:Point` specified in the `kml:Placemark`. The origin of the image coordinate system is in the lower left corner of the icon.

12.12.3.4.2 Content

Type:	kml:vec2Type
Default Value:	See kml:vec2Type

12.12.3.5 kml:IconStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.12.3.6 kml:IconStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.12.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
   <Style id="randomColorIcon">
      <IconStyle>
         <color>ff00ff00</color>
         <colorMode>random</colorMode>
         <scale>1.1</scale>
         <Icon>
            <href>http://maps.google.com/mapfiles/kml/pal3/icon21.png</href>
         </Icon>
      </IconStyle>
   </Style>
   <Placemark>
      <name>IconStyle.kml</name>
      <styleUrl>#randomColorIcon</styleUrl>
      <Point>
         <coordinates>-122.36868,37.831145,0</coordinates>
      </Point>
   </Placemark>
</Document>
</kml>
```

12.13 kml:Icon (kml:BasicLinkType)

12.13.1 Structure

```
<kml:Icon
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:href>...</kml:href> [0..1]
    <kml:BasicLinkSimpleExtensionGroup>...
     </kml:BasicLinkSimpleExtensionGroup> [0..*]
    <kml:BasicLinkObjectExtensionGroup>...
     </kml:BasicLinkObjectExtensionGroup> [0..*]
</kml:Icon>
```

12.13.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Specifies an icon resource location.

kml:Icon should contain the kml:href child element outside of an update context, that is when not a descendant of kml:Update.

12.13.3 Content

12.13.3.1 kml:href

12.13.3.1.1 Description

Specifies the the resource location as a URL. The URL may contain a fragment component that allows indirect identification of some portion or subset of a resource. As defined in RFC 3986, a fragment identifier is indicated by the presence of a number sign ("#") character and terminated by the end of the URL.

The format and semantics of a fragment identifier is dependent on the media type of the resource. The W3C XPointer framework specifies a standard syntax for referring to fragments of XML resources. An element within a KML resource may be referenced using a shorthand pointer that identifies at most one element; specifically, the kml:AbstractObjectGroup element (if any) that has a matching NCName as the value of the id attribute.

EXAMPLE: `http://www.example.org/path/kml-resource#placemark-1`

12.13.3.1.2 Content

Type:	xsd:string
Default Value:	none

12.13.3.2 kml:BasicLinkSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.13.3.3 kml:BasicLinkObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.14 kml:LabelStyle

12.14.1 Structure

```
<kml:LabelStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:abstractColorMode>...</kml:abstractColorMode> [0..1]
    <kml:AbstractColorStyleSimpleExtensionGroup>...
     </kml:AbstractColorStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractColorStyleObjectExtensionGroup>...
     </kml:AbstractColorStyleObjectExtensionGroup> [0..*]
    <kml:scale>...</kml:scale> [0..1]
    <kml:LabelStyleSimpleExtensionGroup>...
     </kml:LabelStyleSimpleExtensionGroup> [0..*]
    <kml:LabelStyleObjectExtensionGroup>...
     </kml:LabelStyleObjectExtensionGroup> [0..*]
</kml:LabelStyle>
```

12.14.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractColorStyleGroup*

Specifies how the name of a *kml:AbstractFeatureGroup* is drawn in the geographic view. A user-defined color, color mode, and scale for the value of the name can be specified.

kml:LabelStyle should contain at least one child element outside of an update context, that is when not a descendant of kml:Update.

12.14.3 Content

12.14.3.1 kml:scale

12.14.3.1.1 Description

Specifies a scale factor to be applied to the label.

12.14.3.1.2 Content

Type:	xsd: double
Default Value:	1.0

12.14.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
   <Style id="randomLabelColor">
      <LabelStyle>
         <color>ff0000cc</color>
         <colorMode>random</colorMode>
         <scale>1.5</scale>
      </LabelStyle>
   </Style>
   <Placemark>
      <name>LabelStyle.kml</name>
      <styleUrl>#randomLabelColor</styleUrl>
      <Point>
         <coordinates>-122.367375,37.829192,0</coordinates>
      </Point>
   </Placemark>
</Document>
</kml>
```

12.15 kml:LineStyle

12.15.1 Structure

```
<kml:LineStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:abstractColorMode>...</kml:abstractColorMode> [0..1]
    <kml:AbstractColorStyleSimpleExtensionGroup>...
     </kml:AbstractColorStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractColorStyleObjectExtensionGroup>...
     </kml:AbstractColorStyleObjectExtensionGroup> [0..*]
    <kml:width>...</kml:width> [0..1]
    <kml:LineStyleSimpleExtensionGroup>...
     </kml:LineStyleSimpleExtensionGroup> [0..*]
    <kml:LineStyleObjectExtensionGroup>...
     </kml:LineStyleObjectExtensionGroup> [0..*]
</kml:LineStyle>
```

12.15.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractColorStyleGroup*

Specifies the drawing style (color, color mode, and line width) for all line geometry. Line geometry includes the `kml:Polygon` boundaries (`kml:LinearRings`) for which the applicable `kml:PolyStyle` outline element value is 1 or true, and lines connecting extruded `kml:Placemarks` with a `kml:Point` geometry to the ground. Use `kml:LineStyle` to specify the color, color mode, and width of the line. For extruded kml:LineStrings, the line itself uses the current kml:LineStyle, and the extrusion uses the current kml:PolyStyle.

12.15.3 Content

12.15.3.1 kml:width

12.15.3.1.1 Description

Width of the line, in pixels.

12.15.3.1.2 Content

Type:	xsd:double
Default Value:	1.0

12.15.3.2 kml:LineStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.15.3.3 kml:LineStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.15.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>LineStyle.kml</name>
  <open>1</open>
  <Style id="linestyleExample">
    <LineStyle>
      <color>7f0000ff</color>
      <width>4</width>
    </LineStyle>
  </Style>
  <Placemark>
    <name>LineStyle Example</name>
    <styleUrl>#linestyleExample</styleUrl>
    <LineString>
      <extrude>1</extrude>
      <tessellate>1</tessellate>
      <coordinates>
        -122.364383,37.824664,0 -122.364152,37.824322,0
      </coordinates>
    </LineString>
  </Placemark>
</Document>
</kml>
```

12.16 kml:PolyStyle

12.16.1 Structure

```
<kml:PolyStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:color>...</kml:color> [0..1]
    <kml:abstractColorMode>...</kml:abstractColorMode> [0..1]
    <kml:AbstractColorStyleSimpleExtensionGroup>...
     </kml:AbstractColorStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractColorStyleObjectExtensionGroup>...
     </kml:AbstractColorStyleObjectExtensionGroup> [0..*]
    <kml:fill>...</kml:fill> [0..1]
    <kml:outline>...</kml:outline> [0..1]
    <kml:PolyStyleSimpleExtensionGroup>...
     </kml:PolyStyleSimpleExtensionGroup> [0..*]
    <kml:PolyStyleObjectExtensionGroup>...
     </kml:PolyStyleObjectExtensionGroup> [0..*]
</kml:PolyStyle>
```

12.16.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractColorStyleGroup*

Specifies the drawing style for a `kml:Polygon`, including a `kml:Polygon` and the extruded portion of a `kml:Polygon` or `kml:LineString`.

`kml:PolyStyle` should contain at least one child element outside of an update context, that is when not a descendant of `kml:Update`.

12.16.3 Content

12.16.3.1 kml:fill

12.16.3.1.1 Description

Specifies whether to fill the polygon. 1 or true specifies to fill the polygon; 0 or false specifies to not fill the polygon.

12.16.3.1.2 Content

Type:	xsd:boolean
Default Value:	1 or true

12.16.3.2 kml:outline

12.16.3.2.1 Description

Specifies whether to outline the polygon. 1 or true specifies to draw the polygon boundaries; 0 or false specifies to not draw the polygon boundaries.

NOTE: Polygon outlines are styled using the current LineStyle.

12.16.3.2.2 Content

Type:	xsd:boolean
Default Value:	1 or true

12.16.3.3 kml:PolyStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.16.3.4 kml:PolyStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.16.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>PolygonStyle.kml</name>
  <open>1</open>
  <Style id="examplePolyStyle">
    <PolyStyle>
      <color>ff0000cc</color>
      <colorMode>random</colorMode>
    </PolyStyle>
  </Style>
  <Placemark>
    <name>hollow box</name>
    <styleUrl>#examplePolyStyle</styleUrl>
    <Polygon>
      <extrude>1</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.3662784465226,37.81884427772081,30
            -122.3652480684771,37.81926777010555,30
            -122.365640222455,37.81986126286519,30
            -122.36666937925,37.81942987753481,30
            -122.3662784465226,37.81884427772081,30
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.366212593918,37.81897719083808,30
            -122.3654241733188,37.81929450992014,30
            -122.3657048517827,37.81973175302663,30
            -122.3664882465854,37.81940249291773,30
            -122.366212593918,37.81897719083808,30
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
    </Polygon>
  </Placemark>
</Document>
</kml>
```

12.17 kml:ListStyle

12.17.1 Structure

```
<kml:ListStyle
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleSimpleExtensionGroup>...
     </kml:AbstractSubStyleSimpleExtensionGroup> [0..*]
    <kml:AbstractSubStyleObjectExtensionGroup>...
     </kml:AbstractSubStyleObjectExtensionGroup> [0..*]
    <kml:abstractListItemType>...</kml:abstractListItemType> [0..1]
    <kml:bgColor>...</kml:bgColor> [0..1]
    <kml:ItemIcon>...</kml:ItemIcon> [0..*]
    <kml:maxSnippetLines>...</kml:maxSnippetLines> [0..1]
    <kml:ListStyleSimpleExtensionGroup>...
     </kml:ListStyleSimpleExtensionGroup> [0..*]
    <kml:ListStyleObjectExtensionGroup>...
     </kml:ListStyleObjectExtensionGroup> [0..*]
</kml:ListStyle>
```

12.17.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractColorStyleGroup*

Specifies how a *kml:AbstractFeatureGroup* is displayed in the list view.

kml:ListStyle should contain at least one child element outside of an update context, that is when not a descendant of kml:Update.

12.17.3 Content

12.17.3.1 kml:abstractListItemType

12.17.3.1.1 Description

An abstract placeholder for kml:listItemType and an extension point for list item type enumerations. See also 12.18 kml:listItemType and 6.7.2.1 Simple Element Substitution.

12.17.3.1.2 Content

Type:	kml:enumBaseType
Default Value:	none

12.17.3.2 kml:bgColor

See 12.8 kml:bgColor.

12.17.3.3 kml:ItemIcon

Icons used in the list view to reflects the state of a kml:Folder or kml:NetworkLink fetch. Multiple kml:ItemIcon elements may be encoded to associate each state with a unique icon.

See 12.19 kml:ItemIcon.

12.17.3.4 kml:maxSnippetLines

12.17.3.4.1 Description

Specifies the maximum number of lines to display for the *kml:AbstractFeatureGroup* kml:snippet value in the list view.

12.17.3.4.2 Content

Type:	xsd:int
Default Value:	2

12.17.3.5 kml:ListStyleSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.17.3.6 kml:ListStyleObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

12.17.4 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <name>ListStyle.kml</name>
  <open>1</open>
  <Style id="bgColorExample">
    <ListStyle>
      <bgColor>ff336699</bgColor>
    </ListStyle>
  </Style>
  <Style id="checkHideChildrenExample">
    <ListStyle>
      <listItemType>checkHideChildren</listItemType>
    </ListStyle>
  </Style>
  <Style id="radioFolderExample">
    <ListStyle>
      <listItemType>radioFolder</listItemType>
    </ListStyle>
  </Style>
  <Folder>
    <name>ListStyle Examples</name>
    <open>1</open>
    <Folder>
      <name>bgColor example</name>
      <open>1</open>
       <styleUrl>#bgColorExample</styleUrl>
      <Placemark>
        <name>pl1</name>
        <Point>
          <coordinates>-122.362815,37.822931,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl2</name>
        <Point>
          <coordinates>-122.362825,37.822931,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl3</name>
        <Point>
          <coordinates>-122.362835,37.822931,0</coordinates>
        </Point>
      </Placemark>
    </Folder>
    <Folder>
      <name>checkHideChildren example</name>
      <open>1</open>
      <styleUrl>#checkHideChildrenExample</styleUrl>
      <Placemark>
        <name>pl4</name>
        <Point>
          <coordinates>-122.362845,37.822941,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl5</name>
        <Point>
          <coordinates>-122.362855,37.822941,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl6</name>
        <Point>
          <coordinates>-122.362865,37.822941,0</coordinates>
        </Point>
      </Placemark>
    </Folder>
    <Folder>
      <name>radioFolder example</name>
      <open>1</open>
      <styleUrl>#radioFolderExample</styleUrl>
      <Placemark>
        <name>pl7</name>
        <Point>
          <coordinates>-122.362875,37.822951,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl8</name>
        <Point>
          <coordinates>-122.362885,37.822951,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>pl9</name>
        <Point>
          <coordinates>-122.362895,37.822951,0</coordinates>
        </Point>
      </Placemark>
    </Folder>
  </Folder>
</Document>
</kml>
```

12.18 kml:listItemType

12.18.1 Structure

```
<kml:listItemType>...</kml:listItemType>
```

12.18.2 Description

Specifies how a `kml:Folder` and its contents shall be displayed as items in the list view.

12.18.3 Content

Type:	kml:listItemTypeEnumType
Default Value:	**check**

12.19 kml:ItemIcon

12.19.1 Structure

```
<kml:ItemIcon
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:abstractState>...</kml:abstractState> [0..1]
    <kml:href>...</kml:href> [0..1]
    <kml:ItemIconSimpleExtensionGroup>...</kml:ItemIconSimpleExtensionGroup>
[0..*]
    <kml:ItemIconObjectExtensionGroup>...</kml:ItemIconObjectExtensionGroup>
[0..*]
</kml:ItemIcon>
```

12.19.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

Specifies the location (kml:href) for an icon used in the list view to reflect the state (kml:state) of the kml:Folder or kml:NetworkLink to which it is associated.

kml:ItemIcon shall contain the kml:href element outside of an update context, that is when not a descendant of kml:Update.

12.19.3 Content

12.19.3.1 kml:abstractState

12.19.3.1.1 Description

An abstract placeholder for kml:state and an extension point for a list of state enumerations. See also 12.20 kml:state and 6.7.2.1 Simple Element Substitution.

12.19.3.1.2 Content

Type:	kml:enumBaseType
Default Value:	none

12.19.3.2 kml:href

See 12.13.3.1 kml:href.

12.19.3.3 kml:ItemIconSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

12.19.3.4 kml:ItemIconObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

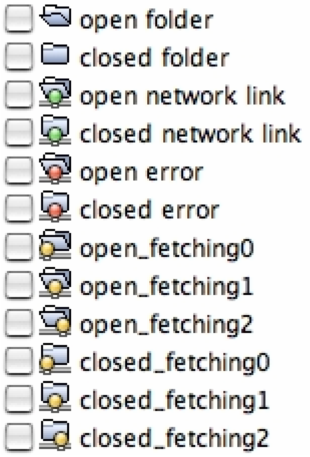
12.20 kml:state

12.20.1 Structure

```
<kml:state>...</kml:state>
```

12.20.2 Description

Specifies the current state of the `kml:NetworkLink` or `kml:Folder`. Icons associated with the **open** and **closed** modes are used for `kml:Folders`. Icons associated with the **error** and **fetching0**, **fetching1**, and **fetching2** modes are used for `kml:NetworkLinks`. Sample icons for each state are shown in the following diagram.



12.20.3 Content

Type:	kml:itemIconStateType
Default Value:	none

13. Links

13.1 kml:Link, kml:Icon (kml:LinkType)

13.1.1 Structure

```
<kml:Link
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:href>...</kml:href> [0..1]
    <kml:BasicLinkSimpleExtensionGroup>...
     </kml:BasicLinkSimpleExtensionGroup> [0..*]
    <kml:BasicLinkObjectExtensionGroup>...
     </kml:BasicLinkObjectExtensionGroup> [0..*]
    <kml:abstractRefreshMode>...</kml:abstractRefreshMode> [0..1]
    <kml:refreshInterval>...</kml:refreshInterval> [0..1]
    <kml:abstractViewRefreshMode>...</kml:abstractViewRefreshMode> [0..1]
    <kml:viewRefreshTime>...</kml:viewRefreshTime> [0..1]
    <kml:viewBoundScale>...</kml:viewBoundScale> [0..1]
    <kml:viewFormat>...</kml:viewFormat> [0..1]
    <kml:httpQuery>...</kml:httpQuery> [0..1]
    <kml:LinkSimpleExtensionGroup>...</kml:LinkSimpleExtensionGroup> [0..*]
    <kml:LinkObjectExtensionGroup>...</kml:LinkObjectExtensionGroup> [0..*]
</kml:Link>
```

```
<kml:Icon
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:href>...</kml:href> [0..1]
    <kml:BasicLinkSimpleExtensionGroup>...
     </kml:BasicLinkSimpleExtensionGroup> [0..*]
    <kml:BasicLinkObjectExtensionGroup>...
     </kml:BasicLinkObjectExtensionGroup> [0..*]
    <kml:abstractRefreshMode>...</kml:abstractRefreshMode> [0..1]
    <kml:refreshInterval>...</kml:refreshInterval> [0..1]
    <kml:abstractViewRefreshMode>...</kml:abstractViewRefreshMode> [0..1]
    <kml:viewRefreshTime>...</kml:viewRefreshTime> [0..1]
    <kml:viewBoundScale>...</kml:viewBoundScale> [0..1]
    <kml:viewFormat>...</kml:viewFormat> [0..1]
    <kml:httpQuery>...</kml:httpQuery> [0..1]
    <kml:LinkSimpleExtensionGroup>...</kml:LinkSimpleExtensionGroup> [0..*]
    <kml:LinkObjectExtensionGroup>...</kml:LinkObjectExtensionGroup> [0..*]
</kml:Icon>
```

13.1.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

NOTE: The schema type for `kml:Link` and `kml:Icon` derives by extension from `kml:BasicLinkType`, whose content model is described in 12.13 kml:Icon (kml:BasicLinkType).

`kml:Link` and `kml:Icon` have the same content model. They both specify the location (`kml:href`) and handling of a resource. For both elements `kml:href` shall be specified when not being used in a feature update context, that is when not a descendant of `kml:Update`.

The resource should be loaded and refreshed according to the refresh parameters supplied. Two different sets of refresh parameters can be specified: one based on time (`kml:refreshMode` and `kml:refreshInterval`) and one based on the current view (`kml:viewRefreshMode` and `kml:viewRefreshTime).`

In addition, `kml:Link` and `kml:Icon` specifies whether to scale any bounding box parameters (`kml:viewBoundScale`), and provides a set of optional viewing parameters (`kml:viewFormat`) as well as a set of optional parameters containing version and language information (`kml:httpQuery`).

The valid URL request shall be the concatenation of three pieces of information:

- the `kml:href` (URL) that specifies the resource location;
- the `kml:viewFormat` string value used to specify any view parameters;
- the `kml:httpQuery` string value used to specify any other query parameters.

An earth browser shall substitute relevant values for query parameters within the URL request.

If `kml:viewFormat` or `kml:httpQuery` are specified, they should contain at least one parameter.

If the resource specified in `kml:href` is a local resource, the `kml:viewFormat` and `kml:httpQuery` elements shall be ignored.

`kml:Link` and `kml:Icon` shall contain the `kml:href` child element outside of an update context, that is when not a descendant of `kml:Update`.

13.1.3 Content

13.1.3.1 kml:abstractRefreshMode

13.1.3.1.1 Description

An abstract placeholder for `kml:refreshMode` and an extension point for refresh mode enumerations. See also 13.2 kml:refreshMode and 6.7.2.1 Simple Element Substitution.

13.1.3.1.2 Content

Type:	kml:enumBaseType
Default Value:	none

13.1.3.2 kml:refreshInterval

13.1.3.2.1 Description

Indicates to refresh the resource every *n* seconds. The value shall be positive.

Content

Type:	xsd:double
Default Value:	4.0

13.1.3.3 kml:abstractViewRefreshMode

13.1.3.3.1 Description

An abstract placeholder for `kml:viewRefreshMode` and an extension point for view refresh mode enumerations. See also 13.3 kml:viewRefreshMode and 6.7.2.1 Simple Element Substitution.

13.1.3.3.2 Content

Type:	kml:enumBaseType
Default Value:	none

13.1.3.4 kml:viewRefreshTime

13.1.3.4.1 Description

Specifies the number of seconds to wait before refreshing the geographic view after camera movement stops. This applies when `kml:viewRefreshMode` is set to **onStop**. The value shall be positive.

13.1.3.4.2 Content

Type:	xsd:double
Default Value:	4.0

13.1.3.5 kml:viewBoundScale

13.1.3.5.1 Description

Scales any bounding box parameters. A value less than 1 specifies the use of a smaller geographic area than the current geographic view. A value greater than 1 specifies the use a larger geographic area greater than the current geographic view. The value shall be positive.

13.1.3.5.2 Content

Type:	xsd:double
Default Value:	1.0

13.1.3.6 kml:viewFormat

13.1.3.6.1 Description

Specifies the format of a query string related to view parameters that is appended to the `kml:href` before the resource is fetched. The following query parameters may be used:

- **[lookatLon]**, **[lookatLat]** – longitude and latitude of the point that `kml:LookAt` is viewing
- **[lookatRange]**, **[lookatTilt]**, **[lookatHeading]** – values used by the `kml:LookAt` element (see descriptions of `kml:range`, `kml:tilt`, and `kml:heading` in `kml:LookAt`)
- **[lookatTerrainLon]**, **[lookatTerrainLat]**, **[lookatTerrainAlt]** – point on the terrain in decimal degrees/meters that `kml:LookAt` is viewing
- **[cameraLon]**, **[cameraLat]**, **[cameraAlt]** – decimal degrees/meters of the eyepoint for the camera
- **[horizFov]**, **[vertFov]** **–** horizontal, vertical field of view for the camera
- **[horizPixels]**, **[vertPixels]** – size in pixels of the geographic view
- **[terrainEnabled]** – indicates whether the geographic view is showing terrain
- **[bboxWest], [bboxSouth], [bboxEast], [bboxNorth]** – bounding box limits matching the OGC Web Map Service (WMS) bounding box specification.

13.1.3.6.2 Content

Type:	xsd:string
Default Value:	none

13.1.3.7 kml:httpQuery

13.1.3.7.1 Description

String value used to specify any additional query parameters not related to the geographic view.

13.1.3.7.2 Content

Type:	xsd:string

Default Value: none

The following query parameters may be used:

- **[clientVersion] –** version of earth browser client
- **[kmlVersion] –** version of requested kml
- **[clientName] –** name of earth browser client
- **[language]** – language preference of the earth browser client

13.1.3.8 kml:LinkSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

13.1.3.9 kml:LinkObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

13.1.4 Example

```
<NetworkLink>
  <name>NE US Radar</name>
  <flyToView>1</flyToView>
  <Link>

<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];CAMERA=\
      [lookatLon],[lookatLat],[lookatRange],[lookatTilt],[lookatHeading];VIEW=\

[horizFov],[vertFov],[horizPixels],[vertPixels],[terrainEnabled]</viewFormat>
  </Link>
</NetworkLink>
```

13.2 kml:refreshMode

13.2.1 Structure

```
<kml:refreshMode>...</kml:refreshMode>
```

13.2.2 Description

Specifies a time-based refresh mode.

If a fetched resource has a `kml:NetworkLinkControl`, the `expires` time takes precedence over expiration times specified in HTTP headers.

If no `expires` time is specified, the HTTP *max-age* header is used (if present). If *max-age* is not present, the *Expires* HTTP header is used (if present). See RFC 2616.

If `kml:refreshInterval` is specified then `kml:refreshMode` should be set to **onInterval**. If `kml:viewRefreshTime` is specified then `kml:refreshMode` should be set to **onStop**.

13.2.3 Content

Type:	kml:refreshModeEnumType
Default Value:	**onChange**

13.3 kml:viewRefreshMode

13.3.1 Structure

```
<kml:viewRefreshMode>...</kml:viewRefreshMode>
```

13.3.2 Description

Specifies how the link is refreshed when the geographic view changes.

13.3.3 Content

Type:	kml:viewRefreshModeEnumType
Default Value:	**never**

13.4 kml:NetworkLinkControl

13.4.1 Structure

```
<kml:NetworkLinkControl
 anyAttribute="anySimpleType [0..1]">
    <kml:minRefreshPeriod>...</kml:minRefreshPeriod> [0..1]
    <kml:maxSessionLength>...</kml:maxSessionLength> [0..1]
    <kml:cookie>...</kml:cookie> [0..1]
    <kml:message>...</kml:message> [0..1]
    <kml:linkName>...</kml:linkName> [0..1]
    <kml:linkDescription>...</kml:linkDescription> [0..1]
    <kml:linkSnippet>...</kml:linkSnippet> [0..1]
    <kml:expires>...</kml:expires> [0..1]
    <kml:Update>...</kml:Update> [0..1]
    <kml:AbstractViewGroup>...</kml:AbstractViewGroup> [0..1]
    <kml:NetworkLinkControlSimpleExtensionGroup>...
     </kml:NetworkLinkControlSimpleExtensionGroup> [0..*]
    <kml:NetworkLinkControlObjectExtensionGroup>...
     </kml:NetworkLinkControlObjectExtensionGroup> [0..*]
</kml:NetworkLinkControl>
```

13.4.2 Description

Controls the behaviour of a `kml:NetworkLink` that references the KML resource to which the *`kml:NetworkLinkControl`* belongs.

See also 9.15 kml:NetworkLink.

13.4.3 Content

13.4.3.1 kml:minRefreshPeriod

13.4.3.1.1 Description

Specifies in seconds the minimum allowed time between refreshes of the referenced KML resource. The value shall take precedence over the `kml:refreshInterval` element value specified by the `kml:NetworkLink`.

13.4.3.1.2 Content

Type:	xsd:double
Default Value:	0.0

13.4.3.2 kml:maxSessionLength

13.4.3.2.1 Description

Specifies in seconds the maximum time that an earth browser shall remain connected to the referenced KML resource. The default value of -1 indicates not to terminate the session explicitly.

13.4.3.2.2 Content

Type:	xsd: double
Default Value:	-1.0

13.4.3.3 kml:cookie

13.4.3.3.1 Description

Use this element to append a string to the `kml:NetworkLink` URL query.

13.4.3.3.2 Content

Type:	xsd:string
Default Value:	none

13.4.3.4 kml:message

13.4.3.4.1 Description

Text that should be displayed when a `kml:NetworkLink` is first activated or the `kml:message` value is updated.

13.4.3.4.2 Content

Type:	xsd:string

Default Value: none

13.4.3.5 kml:linkName

13.4.3.5.1 Description

Specifies valid content for the `kml:NetworkLink` `kml:name` element. The `kml:linkName` value shall take precedence over the value of `kml:name` value.

13.4.3.5.2 Content

Type:	xsd:string
Default Value:	none

13.4.3.6 kml:linkDescription

13.4.3.6.1 Description

Specifies text for the `kml:NetworkLink` `kml:description` element. The `kml:linkDescription` value shall take precedence over the `kml:NetworkLink` `kml:description` value.

The text may include HTML content that is encoded as well-formed XML using HTML entity references or by enclosing the HTML within a CDATA section.

13.4.3.6.2 Content

Type:	xsd:string
Default Value:	none

13.4.3.7 kml:linkSnippet

13.4.3.7.1 Description

Specifies overriding text for the kml:snippet child element value of the `kml:NetworkLink` element. The `kml:linkSnippet` content shall take precedence over the `kml:NetworkLink/kml:snippet` value (and likewise for the deprecated kml:NetworkLink/kml:Snippet value if present).The text may include HTML content that is encoded as well-formed XML using HTML entity references or by enclosing the HTML within a CDATA section.

See also 9.1.3.10 kml:snippet.

13.4.3.7.2 Content

Type:	xsd:string
Default Value:	none

13.4.3.8 kml:expires

13.4.3.8.1 Description

Specifies a point in time at which the `kml:NetworkLink` shall be refreshed. It applies only if an associated `kml:Link refreshMode` *value* is **onExpire**.

13.4.3.8.2 Content

Type:	kml:dateTimeType
Default Value:	none

13.4.3.9 kml:Update

See 13.5 kml:Update.

13.4.3.10 kml:AbstractViewGroup

See 14.1 AbstractViewGroup.

13.4.3.11 kml:NetworkLinkControlSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

13.4.3.12 kml:NetworkLinkControlObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

13.4.4 Attributes

13.4.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

13.4.5 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLinkControl>
     <message>This is a pop-up message. You will only see this once</message>
     <cookie>cookie=sometext</cookie>
     <linkName>New KML features</linkName>
     <linkDescription><![CDATA[KML now has new features
                         available!]]></linkDescription>
  </NetworkLinkControl>
</kml>
```

13.5 kml:Update

13.5.1 Structure

```
<kml:Update
 anyAttribute="anySimpleType [0..1]">
    <kml:targetHref>...</kml:targetHref> [1]
    <kml:AbstractUpdateOptionGroup>...</kml:AbstractUpdateOptionGroup> [1..*]
    <kml:UpdateExtensionGroup>...</kml:UpdateExtensionGroup> [0..*]
</kml:Update>
```

13.5.2 Description

Specifies an addition, change, or deletion to a KML resource that has previously been retrieved via `kml:NetworkLink`.

NOTE: Update does not affect the KML resource itself; rather it updates its representation within the earth browser only.

All KML objects within an update context, that is a grandchild of the `kml:Update` element shall include a `targetId` attribute that identifies the object to be updated, and shall not have an `id` attribute.

`kml:Update` shall have at least one child element.

13.5.3 Content

13.5.3.1 kml:targetHref

13.5.3.1.1 Description

Specifies the URL for the target KML resource that has been previously retrieved via `kml:NetworkLink`.

13.5.3.1.2 Content

Type:	xsd:anyURI
Default Value:	none

13.5.3.2 kml:AbstractUpdateOptionGroup

13.5.3.2.1 Description

An abstract placeholder for `kml:Create`, `kml:Delete`, or `kml:Change`.

See 13.6 kml:Create, 13.7 kml:Delete, 13.8 kml:Change, and 6.7.2.2 Complex Element Substitution.

13.5.3.2.2 Content

Type:	xsd:anyType

13.5.3.3 kml:UpdateExtensionGroup

See 6.7.2.2 Complex Element Substitution.

13.5.4 Attributes

13.5.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

13.6 kml:Create

13.6.1 Structure

```
<kml:Create
 anyAttribute="anySimpleType [0..1]">
   <kml:AbstractContainerGroup>...</kml:AbstractContainerGroup> [0..*]
   <kml:MultiTrack>...</kml:MultiTrack> [0..*]
   <kml:MultiGeometry>...</kml:MultiGeometry> [0..*]
</kml:Create>
```

13.6.2 Description

Specifies the addition of zero or more *`kml:AbstractFeatureGroup`*, `kml:Track`, or *`kml:AbstractGeometryGroup`* elements to an appropriate container element, i.e. `kml:Folder`, `kml:Document` , `kml:MultiTrack`, or `kml:MultiGeometry` in the target resource.

The container element shall be identified as a child of the `kml:Create` element and shall include the `kml:targetId` attribute, and omit the `id` attribute. New *`kml:AbstractFeatureGroup`*, `kml:Track`, or *`kml:AbstractGeometryGroup`* elements to be added to the identified container element are specified as the content of this child.

The `kml:targetHref` for a created *`kml:AbstractFeatureGroup`*, `kml:Track`, or *`kml:AbstractGeometryGroup`* is the same as that of the target KML resource.

13.6.3 Content

13.6.3.1 kml:AbstractContainerGroup

See 9.8 kml:AbstractContainerGroup.

13.6.3.2 kml:MultiTrack

See 10.16 kml:MultiTrack.

13.6.3.3 kml:MultiGeometry

See 10.2 kml:MultiGeometry.

13.6.4 Attributes

13.6.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

13.6.5 Example

This example creates a new `kml:Placemark` in a previously created `kml:Document` that has an `id` of `region24`. Note that to make subsequent updates to `placemark891`, `http://myserver.com/Point.kml` is used as the `kml:targetHref` value.

```
<Update>
  <targetHref>http://myserver.com/Point.kml</targetHref>
  <Create>
    <Document targetId="region24">
      <Placemark id="placemark891">
        <Point>
          <coordinates>-95.48,40.43,0</coordinates>
        </Point>
      </Placemark>
    </Document>
  </Create>
</Update>
```

13.7 kml:Delete

13.7.1 Structure

```
<kml:Delete
 anyAttribute="anySimpleType [0..1]">
    <kml:AbstractFeatureGroup>...</kml:AbstractFeatureGroup> [0..*]
    <kml:AbstractGeometryGroup>...</kml:AbstractGeometryGroup> [0..*]
</kml:Delete>
```

13.7.2 Description

Specifies the deletion of zero or more *`kml:AbstractFeatureGroup`* or *`kml:AbstractGeometryGroup`* elements in the target resource.

Features or Geometries to be deleted shall be identified as children of the `kml:Delete` element and shall include the `kml:targetId` attribute and omit the `id` attribute.

13.7.3 Content

13.7.3.1 kml:AbstractFeatureGroup

See 9.1 kml:AbstractFeatureGroup.

13.7.3.2 kml:AbstractGeometryGroup

See 10.1 kml:AbstractGeometryGroup.

13.7.4 Attributes

13.7.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

13.7.5 Example

This example deletes a `kml:Placemark` previously loaded into an earth browser. This `kml:Placemark` may have been loaded directly by a `kml:NetworkLink` with the specified URL, or it may have been loaded by a subsequent `kml:Update` to the original `kml:Document`.

```
<Update>
  <targetHref>http://www.foo.com/Point.kml</targetHref>
  <Delete>
    <Placemark targetId="pa3556"></>
  </Delete>
</Update>
```

13.8 kml:Change

13.8.1 Structure

```
<kml:Change
 anyAttribute="anySimpleType [0..1]">
    <kml:AbstractObjectGroup>...</kml:AbstractObjectGroup> [0..*]
</kml:Change>
```

13.8.2 Description

Specifies modifications to zero or more identified *`kml:AbstractObjectGroup`* elements in the target resource.

Target elements to be modified are identified as children of the `kml:Change` element and shall include the `kml:targetId` attribute and omit the `id` attribute. Modifications to the identified *`kml:AbstractObjectGroup`* are specified by the content of these children.

The content of identified target elements not subject to modification shall remain unchanged.

13.8.3 Content

13.8.3.1 kml:AbstractObjectGroup

See 8.1 kml:AbstractObjectGroup.

13.8.3.2 kml:NetworkLinkControlSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

13.8.3.3 kml:NetworkLinkControlObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

13.8.4 Attributes

13.8.4.1 anyAttribute

See 7.1.4.3 anyAttribute.

13.8.5 Example

```
<NetworkLinkControl>
  <Update>
    <targetHref>http://www/~sam/January14Data/Point.kml</targetHref>
    <Change>
      <Point targetId="point123">
        <coordinates>-95.48,40.43,0</coordinates>
      </Point>
    </Change>
  </Update>
</NetworkLinkControl>
```

14. Views

14.1 kml:AbstractViewGroup

14.1.1 Structure

```
<kml:AbstractViewGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:AbstractViewSimpleExtensionGroup>...
     </kml:AbstractViewSimpleExtensionGroup> [0..*]
    <kml:AbstractViewObjectExtensionGroup>...
     </kml:AbstractViewObjectExtensionGroup> [0..*]
</kml:AbstractViewGroup>
```

14.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:LookAt
- kml:Camera

An earth browser displays KML graphics overlaid on a backdrop image which is typically that of the earth. In addition to describing the overlaid graphic elements, KML can define a geographic view of the overlaid graphics and the backdrop image.

14.1.3 Content

14.1.3.1 kml:AbstractTimePrimitiveGroup

Inclusion of time information in kml:AbstractView enables flying to a specific location in space and time. The specified time is used to decide which imagery, features, and sunlight to display.

See 15.1 kml:AbstractTimePrimitiveGroup.

14.1.3.2 kml:AbstractViewSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

14.1.3.3 kml:AbstractViewObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

14.2 kml:Camera

14.2.1 Structure

```
<kml:Camera
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:AbstractViewSimpleExtensionGroup>...
     </kml:AbstractViewSimpleExtensionGroup> [0..*]
    <kml:AbstractViewObjectExtensionGroup>...
     </kml:AbstractViewObjectExtensionGroup> [0..*]
    <kml:longitude>...</kml:longitude> [0..1]
    <kml:latitude>...</kml:latitude> [0..1]
    <kml:altitude>...</kml:altitude> [0..1]
    <kml:heading>...</kml:heading> [0..1]
    <kml:tilt>...</kml:tilt> [0..1]
    <kml:roll>...</kml:roll> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:horizFov>...</kml:horizFov> [0..1]
    <kml:CameraSimpleExtensionGroup>...</kml:CameraSimpleExtensionGroup> [0..*]
    <kml:CameraObjectExtensionGroup>...</kml:CameraObjectExtensionGroup> [0..*]
</kml:Camera>
```

14.2.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractViewGroup*

The kml:Camera element specifies the position and orientation of a virtual camera. This can be used to specify views of the earth or of objects in space.

A kml:Camera element should contain the kml:longitude, kml:latitude, and kml:altitude child elements outside of an update context, that is when not a descendant of kml:Update.

14.2.3 Defining a View

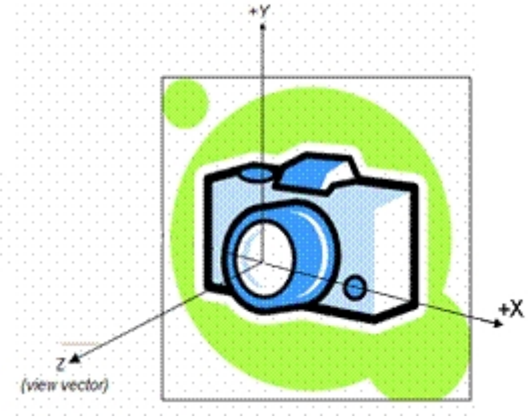
The kml:Camera element specifies the position of the view point of the camera using the child elements kml:longitude, kml:latitude, kml:altitude and kml:altitudeMode. The orientation of the camera is specified using the additional child elements kml:heading, kml:tilt and kml:roll.

The initial or zero position of the camera is defined by an earth-fixed frame lying in a meridian plane (plane containing the view point, the poles, and the earth's center of mass), with the Z'-axis normal to the earth's surface, the Y'-axis directed away from the equator, and the X'-axis such as to form a right handed orthogonal frame.

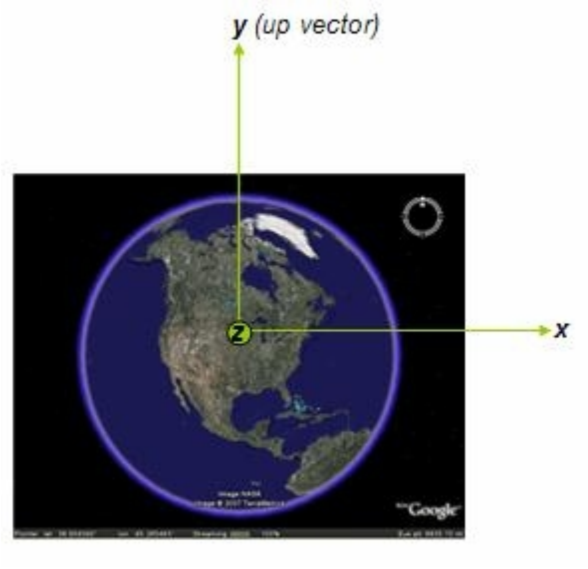
A body-fixed reference frame is assumed attached to the virtual camera, with the Z-axis along the optical axis of the camera, the Y-axis through the top of the camera, and the X-axis such as to form a right handed orthogonal frame. In the zero position of the camera, the camera body Z-axis is aligned with the –Z' axis and the body Y axis with the Y' axis. The orientation of the camera is then defined by the following sequence of rotations (Euler angles) which must be performed in the stated order:

- `kml:altitude` – translate along the Z' axis to the specified altitude.
- `kml:heading` – clockwise rotation around the Z axis. The range of the heading is from 0 to 360 degrees.
- `kml:tilt` – counter clockwise rotation around the X axis. The range of the tilt is from -180 to + 180 degrees.
- `kml:roll` – clockwise around the Z axis (again). The range of the roll is from -180 to +180 degrees

The camera body axes are shown as follows:



The earth-fixed frame specifying the initial (zero) orientation of the camera is illustrated below:



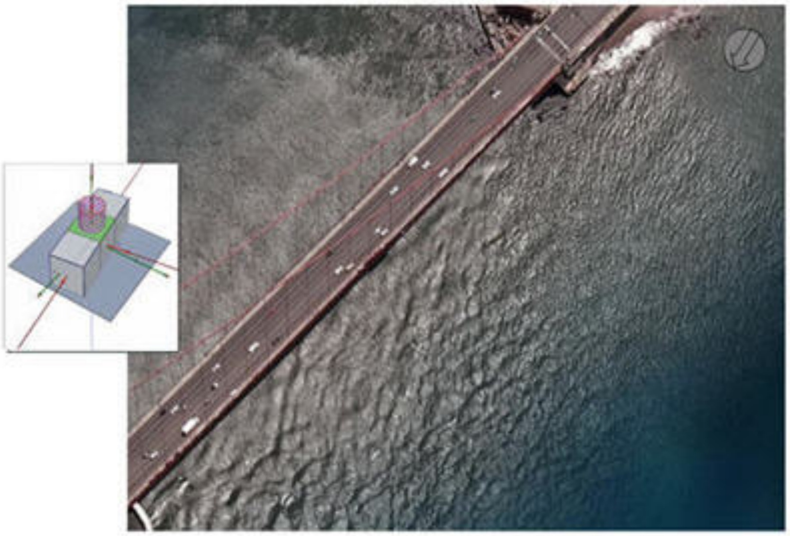
14.2.4 Order of Rotation

The order of rotation is important. By default, the camera shall look straight down the Z axis towards the Earth. The order of rotation is:

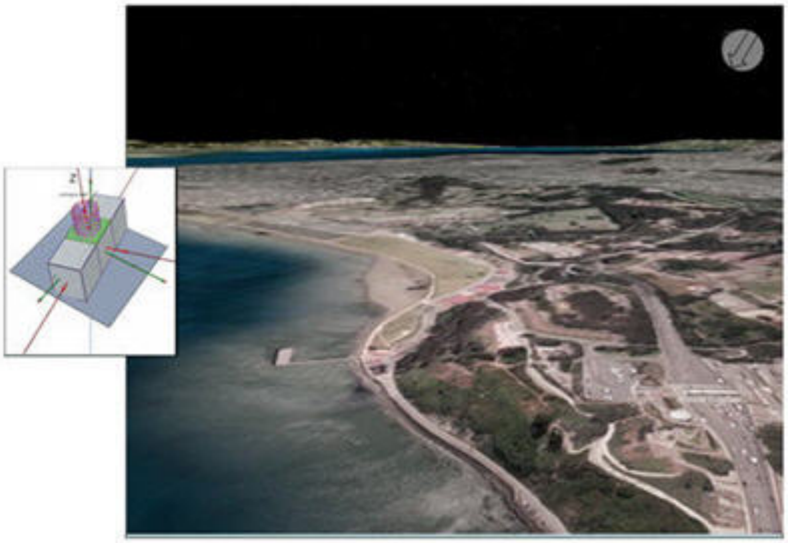
1. `kml:heading` – rotate around the Z axis.
5. `kml:tilt` – rotate around the X axis.
6. `kml:roll` – rotate around the Z axis (again).

The camera's view direction is a vector that is computed from these three rotations. Note that each time a rotation is applied, two of the camera axes shall change their orientation.

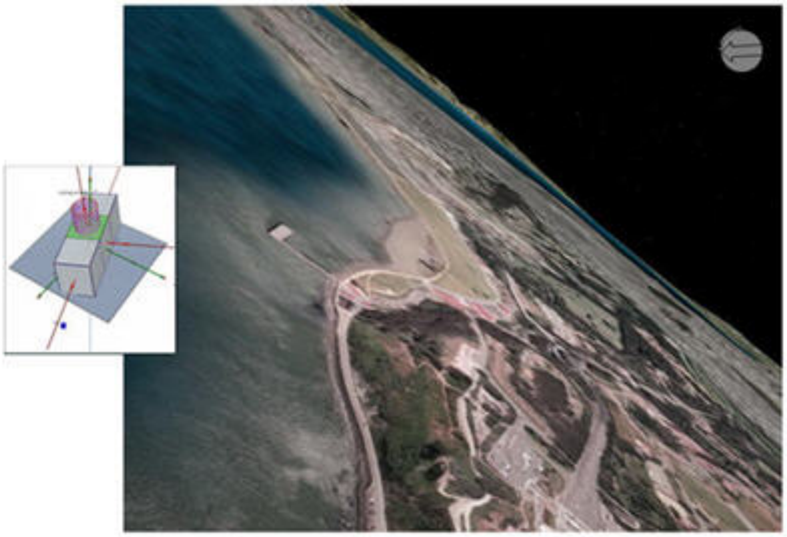
Heading (Rotation about Z):



Tilt (Rotation about X):



Roll (Rotation about Z again)



14.2.5 Content

14.2.5.1 kml:longitude

14.2.5.1.1 Description

Geodetic longitude of the virtual camera (eye point). Angular distance in decimal degrees, relative to the Prime Meridian. Values west of the Meridian range from −180 to 0 degrees. Values east of the Meridian range from 0 to 180 degrees.

14.2.5.1.2 Content

Type:	kml:angle180Type
Default Value:	0.0

14.2.5.2 kml:latitude

14.2.5.2.1 Description

Geodetic latitude of the virtual camera. Decimal degrees north or south of the Equator (0 degrees). Values range from −90 degrees to 90 degrees.

14.2.5.2.2 Content

Type:	kml:angle90Type
Default Value:	0.0

14.2.5.3 kml:altitude

14.2.5.3.1 Description

Distance of the camera from the Earth's surface, in meters, interpreted according to the `kml:altitudeMode` specification.

14.2.5.3.2 Content

Type:	xsd:double
Default Value:	0.0

14.2.5.4 kml:heading

14.2.5.4.1 Description

Direction (azimuth) of the camera, in decimal degrees. Values range from 0 (North) to 360 degrees.

14.2.5.4.2 Content

Type:	kml:angle360Type
Default Value:	0.0

14.2.5.5 kml:tilt

14.2.5.5.1 Description

Rotation, in decimal degrees, of the camera around the X axis. A value of 0 indicates that the view is aimed straight down toward the earth (the most common case). A value of 90 for `kml:tilt` indicates that the view is aimed toward the horizon. Values greater than 90 indicate that the view is pointed up into the sky. Values for `kml:tilt` are clamped at +180 degrees.

14.2.5.5.2 Content

Type:	kml:angle180Type
Default Value:	0.0

14.2.5.6 kml:roll

14.2.5.6.1 Description

Rotation, in decimal degrees, of the camera around the Z axis. Values range from −180 to +180 degrees.

14.2.5.6.2 Content

Type:	kml:angle180Type
Default Value:	0.0

14.2.5.7 kml:altitudeMode

See 9.20 kml:altitudeMode.

14.2.5.8 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

14.2.5.9 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

14.2.5.10 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

14.2.5.11 kml:horizFov

14.2.5.11.1 Description

Defines the horizontal field of view of the `kml:Camera` in the context of `kml:Tour`. This element has no effect on `kml:Camera` outside of the `kml:Tour` context.

14.2.5.11.2 Content

Type:	kml:angle180Type
Default Value:	none

14.2.5.12 kml:CameraSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

14.2.5.13 kml:CameraObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

14.3 kml:LookAt

14.3.1 Structure

```
<kml:LookAt
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveGroup>...</kml:AbstractTimePrimitiveGroup> [0..1]
    <kml:AbstractViewSimpleExtensionGroup>...
     </kml:AbstractViewSimpleExtensionGroup> [0..*]
    <kml:AbstractViewObjectExtensionGroup>...
     </kml:AbstractViewObjectExtensionGroup> [0..*]
    <kml:longitude>...</kml:longitude> [0..1]
    <kml:latitude>...</kml:latitude> [0..1]
    <kml:altitude>...</kml:altitude> [0..1]
    <kml:heading>...</kml:heading> [0..1]
    <kml:tilt>...</kml:tilt> [0..1]
    <kml:range>...</kml:range> [0..1]
    <kml:altitudeMode>...</kml:altitudeModeGroup> [0..1]
    <kml:seaFloorAltitudeMode>...</kml:seaFloorAltitudeMode> [0..1]
    <kml:AltitudeModeSimpleExtensionGroup>...
     </kml:AltitudeModeSimpleExtensionGroup> [0..1]
    <kml:AltitudeModeObjectExtensionGroup>...
     </kml:AltitudeModeObjectExtensionGroup> [0..1]
    <kml:horizFov>...</kml:horizFov> [0..1]
    <kml:LookAtSimpleExtensionGroup>...</kml:LookAtSimpleExtensionGroup> [0..*]
    <kml:LookAtObjectExtensionGroup>...</kml:LookAtObjectExtensionGroup> [0..*]
</kml:LookAt>
```

14.3.2 Description

This element can be used wherever the following element is referenced:

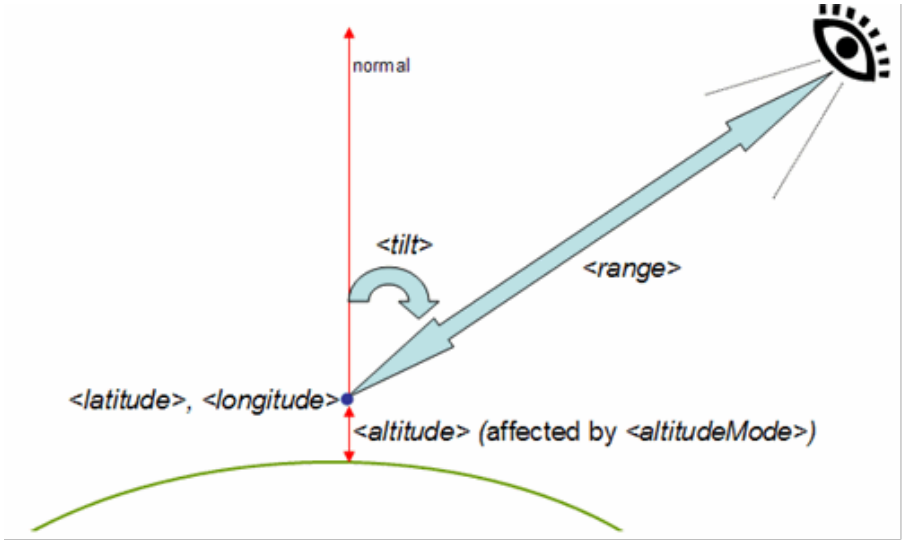
- *kml:AbstractViewGroup*

Specifies the geographic view in terms of a point of interest viewed from a virtual camera. The kml:LookAt object is more limited in scope than kml:Camera and should establish a view direction that intersects the Earth's surface.

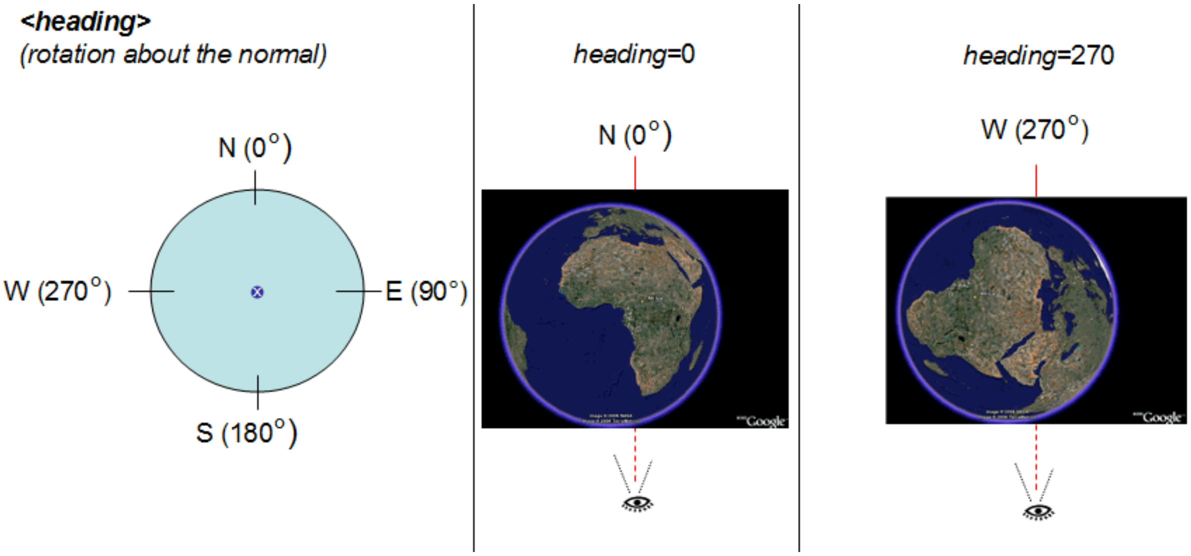
A kml:LookAt element shall contain the kml:longitude, kml:latitude, and kml:range child elements outside of an update context, that is when not a descendant of kml:Update.

14.3.3 Defining How to Look

This diagram illustrates the `kml:range`, `kml:tilt`, and `kml:altitude` elements:



This diagram illustrates the `kml:heading` element:



14.3.4 Content

14.3.4.1 kml:longitude

14.3.4.1.1 Description

Geodetic longitude of the point the camera is looking at. Angular distance in decimal degrees, relative to the Prime Meridian. Values west of the Meridian range from −180 to 0 degrees. Values east of the Meridian range from 0 to 180 degrees.

14.3.4.1.2 Content

Type:	kml:angle180Type
Default Value:	0.0

14.3.4.2 kml:latitude

14.3.4.2.1 Description

Geodetic latitude of the point the camera is looking at. Decimal degrees north or south of the Equator (0 degrees). Values range from −90 degrees to 90 degrees.

14.3.4.2.2 Content

Type:	kml:angle90Type
Default Value:	0.0

14.3.4.3 kml:altitude

14.3.4.3.1 Description

Altitude in meters, interpreted according to the `kml:altitudeMode` specification.

14.3.4.3.2 Content

Type:	xsd:double
Default Value:	0.0

14.3.4.4 kml:heading

14.3.4.4.1 Description

Direction (North, South, East, West), in decimal degrees. Values range from 0 (North) to 360 degrees.

14.3.4.4.2 Content

Type:	kml:angle360Type
Default Value:	0.0

14.3.4.5 kml:tilt

14.3.4.5.1 Description

Angle, in decimal degrees, between the direction of the `LookAt` position and the normal to the surface of the Earth. Values range from 0 to 90 degrees. Values for `kml:tilt` cannot be negative. A `kml:tilt` value of 0 degrees indicates viewing from directly above. A `kml:tilt` value of 90 degrees indicates viewing along the horizon.

14.3.4.5.2 Content

See 16.6 kml:anglepos180Type.

14.3.4.6 kml:range

14.3.4.6.1 Description

Distance in meters from the point specified by `kml:longitude`, `kml:latitude`, and `kml:altitude` to the `kml:LookAt` position.

14.3.4.6.2 Content

Type:	xsd:double
Default Value:	0.0

14.3.4.7 kml:altitudeMode

See 9.20 kml:altitudeMode.

14.3.4.8 kml:seaFlooraltitudeMode

See 9.21 kml:seaFloorAltitudeMode.

14.3.4.9 kml:AltitudeModeSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

14.3.4.10 kml:AltitudeModeObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

14.3.4.11 kml:horizFov

14.3.4.11.1 Description

Defines the horizontal field of view of `kml:LookAt` in the context of `kml:Tour`. This element has no effect on `kml:LookAt` outside of the `kml:Tour` context.

14.3.4.11.2 Content

Type: kml:angle180Type
Default Value: none

14.3.4.12 kml:LookAtSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

14.3.4.13 kml:LookAtObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

14.3.5 Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
  <name>LookAt.kml</name>
  <LookAt>
    <longitude>-122.363</longitude>
    <latitude>37.81</latitude>
    <altitude>2000</altitude>
    <range>500</range>
    <tilt>45</tilt>
    <heading>0</heading>
    <altitudeMode>relativeToGround</altitudeMode>
  </LookAt>
  <Point>
    <coordinates>-122.363,37.82,0</coordinates>
  </Point>
</Placemark>
</kml>
```

15. Time

15.1 kml:AbstractTimePrimitiveGroup

15.1.1 Structure

```
<kml:AbstractTimePrimitiveGroup
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveSimpleExtensionGroup>...
     </kml:AbstractTimePrimitiveSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveObjectExtensionGroup>...
     </kml:AbstractTimePrimitiveObjectExtensionGroup> [0..*]
</kml:AbstractTimePrimitiveGroup>
```

15.1.2 Description

This abstract element can be used wherever the following element is referenced:

- *kml:AbstractObjectGroup*

The following elements can be used wherever this abstract element is referenced:

- kml:TimeStamp
- kml:TimeSpan

Time values encoded within elements that extend *kml:AbstractTimePrimitiveGroup* shall be in the context of the temporal reference system specified by ISO 8601, which uses the Gregorian Calendar and 24 hour local or Coordinated Universal Time (UTC). See also 9.8 kml:AbstractContainerGroup regarding the inheritance of the *kml:AbstractTimePrimitiveGroup* within KML feature hierarchies.

15.1.3 Content

15.1.3.1 kml:AbstractTimePrimitiveSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

15.1.3.2 kml:AbstractTimePrimitiveObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

15.2 kml:TimeSpan

15.2.1 Structure

```
<kml:TimeSpan
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveSimpleExtensionGroup>...
     </kml:AbstractTimePrimitiveSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveObjectExtensionGroup>...
     </kml:AbstractTimePrimitiveObjectExtensionGroup> [0..*]
    <kml:begin>...</kml:begin> [0..1]
    <kml:end>...</kml:end> [0..1]
    <kml:TimeSpanSimpleExtensionGroup>...
     </kml:TimeSpanSimpleExtensionGroup> [0..*]
    <kml:TimeSpanObjectExtensionGroup>...
     </kml:TimeSpanObjectExtensionGroup> [0..*]
</kml:TimeSpan>
```

15.2.2 Description

This element can be used wherever the following element is referenced:

- *kml:AbstractTimePrimitiveGroup*

Specifies an extent in time bounded by begin and end temporal values. At least one of the child elements `kml:begin` and `kml:end` shall be encoded.

15.2.3 Content

15.2.3.1 kml:begin

15.2.3.1.1 Description

Describes the beginning instant of a time period. If absent, the beginning of the period is unbounded. The value shall be encoding according to the `kml:dateTimeType` field type.

15.2.3.1.2 Content

Type:	kml:dateTimeType
Default Value:	none

15.2.3.2 kml:end

15.2.3.2.1 Description

Describes the ending instant of a time period. If absent, the end of the period is unbounded. The value shall be later than the `kml:begin` value. The value shall be encoding according to the `kml:dateTimeType` field type.

15.2.3.2.2 Content

Type: kml:dateTimeType
Default Value: none

15.2.3.3 kml:TimeSpanSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

15.2.3.4 kml:TimeSpanObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

15.2.4 Example

The following example shows the time period representing Colorado's statehood. It contains only a `kml:begin` element because Colorado became a state on August 1, 1876, and continues to be a state:

```
<Placemark>
  <name>Colorado</name>
  ...
  <TimeSpan>
    <begin>1876-08-01</begin>
  </TimeSpan>
</Placemark>
```

15.3 kml:TimeStamp

15.3.1 Structure

```
<kml:TimeStamp
 id="ID [0..1]"
 targetId="NCName [0..1]"
 anyAttribute="anySimpleType [0..1]">
    <kml:ObjectSimpleExtensionGroup>...</kml:ObjectSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveSimpleExtensionGroup>...
     </kml:AbstractTimePrimitiveSimpleExtensionGroup> [0..*]
    <kml:AbstractTimePrimitiveObjectExtensionGroup>...
     </kml:AbstractTimePrimitiveObjectExtensionGroup> [0..*]
    <kml:when>...</kml:when> [0..1]
    <kml:TimeStampSimpleExtensionGroup>...
     </kml:TimeStampSimpleExtensionGroup> [0..*]
    <kml:TimeStampObjectExtensionGroup>...
     </kml:TimeStampObjectExtensionGroup> [0..*]
</kml:TimeStamp>
```

15.3.2 Description

This element can be used wherever the following element is referenced:

- *`kml:AbstractTimePrimitiveGroup`*

Specifies a single moment in time within the required `kml:when` child element.

15.3.3 Content

15.3.3.1 kml:when

15.3.3.1.1 Description

Specifies a single moment in time.

15.3.3.1.2 Content

Type: kml:dateTimeType
Default Value: none

15.3.3.1.3 Examples

The following examples show different temporal resolutions for the `kml:when` value:

- xsd:gYear *(*YYYY*)*

```
<TimeStamp>
  <when>1997</when>
</TimeStamp>
```

- xsd:gYearMonth *(*YYYY-MM*)*

```
<TimeStamp>
  <when>1997-07</when>
</TimeStamp>
```

- xsd:date *(*YYYY-MM-DD*)*

```
<TimeStamp>
  <when>1997-07-16</when>
</TimeStamp>
```

- xsd:dateTime *(*YYYY-MM-DD***T***hh:mm:ss***Z)***
  Here, T is the separator between the calendar and the hourly notation of time, and Z indicates UTC. (Seconds are required.)

```
<TimeStamp>
  <when>1997-07-16T07:30:15Z</when>
</TimeStamp>
```

- xsd:dateTime *(*YYYY-MM-DD***T***hh:mm:sszzzzzz*)*
  This example gives the local time and then the ± conversion to UTC.

```
<TimeStamp>
  <when>1997-07-16T10:30:15+03:00</when>
</TimeStamp>
```

15.3.3.2 kml:TimeStampSimpleExtensionGroup

See 6.7.2.1 Simple Element Substitution.

15.3.3.3 kml:TimeStampObjectExtensionGroup

See 6.7.2.2 Complex Element Substitution.

16. Field Types

Most of the field types described in this section are simple types, i.e. do not have child elements or attributes, with the exception of the deprecated kml:SnippetType. All of the field types either derive from, or directly use, one or more of the following XML Schema Definition (XSD) datatypes:

```
xsd:date      xsd:dateTime    xsd:double    xsd:float
xsd:gYear     xsd:gYearMonth  xsd:hexBinary xsd:int
xsd:string    xsd:boolean
```

For all KML field types based on float or double, the radix point shall be encoded as '.', i.e. the full stop or period symbol (ASCII code 0x2E). Note that we use the more general term 'radix point' instead of 'decimal point' because float and double are base 2 number representations (with 32 and 64 bit precision, respectively).

16.1 kml:altitudeModeEnumType

16.1.1 Content

Base XSD Type: xsd:string
value comes from list: {'clampToGround'|'relativeToGround'|'absolute'}

clampToGround The specified altitude value will be overridden and set to the terrain surface instead.

relativeToGround Interpret the altitude in meters relative to the terrain surface elevation.

absolute Interpret the altitude as a value in meters relative to the vertical datum.

16.2 kml:seaFloorAltitudeModeEnumType

16.2.1 Content

Base XSD Type: xsd:string
value comes from list: {'clampToSeaFloor'|'relativeToSeaFloor'}

clampToSeaFloor Specified altitude value will be overridden and set to the sea floor instead.

relativeToSeaFloor Interpret the altitude in meters relative to the sea floor.

16.3 kml:angle180Type

16.3.1 Content

Base XSD Type: xsd:double
-180.0 <= value <= 180.0

16.4 kml:angle360Type

16.4.1 Content

Base XSD Type: xsd:double
-360.0 <= value <= 360.0

16.5 kml:angle90Type

16.5.1 Content

Base XSD Type: xsd:double
-90.0 <= value <= 90.0

16.6 kml:anglepos180Type

16.6.1 Content

Base XSD Type: xsd:double
0.0 <= value <= 180.0

16.7 kml:anglepos90Type

16.7.1 Content

Base XSD Type: xsd:double
0.0 <= value <= 90.0

16.8 kml:colorModeEnumType

16.8.1 Description

Specifies the color mode for a graphic element.

Values for <colorMode> are normal (no effect) and random. A value of random applies a random linear scale to the base <color> as follows:

To achieve a truly random selection of colors, specify a base <color> of transparent white (00ffffff).

If a single color component is specified (for example, a value of ff0000ff for red), random color values for that one component (red) will be selected. In this case, the values would range from 00 (black) to ff (full red).

If values for two or for all three color components are specified, a random linear scale is applied to each color component, with results ranging from black to the maximum values specified for each component.

The opacity of a color comes from the alpha component of `color` and is never randomized.

16.8.2 Content

Base XSD Type: xsd:string
value comes from list: {'normal'|'random'}

normal	Specifies a single colour value.
random	Specifies to use a random colour value.

16.9 kml:colorType

16.9.1 Description

Specifies the color of a graphic or text element.

Color and opacity (alpha) values are expressed in hexadecimal notation. The range of values for any one color is 0 to 255 (00 to ff); opaque white is ffffffff; opaque black is ff000000.

The order of expression is *aabbggrr*, where *aa=alpha* (00 to ff); *bb=blue* (00 to ff); *gg=green* (00 to ff); *rr=red* (00 to ff).

For alpha, 00 is fully transparent and ff is fully opaque. For example, to apply a blue color with 50 percent opacity to an overlay, specify the following: <bgColor>7fff0000</bgColor>, where *alpha*=0x7f, *blue*=0xff, *green*=0x00, and *red*=0x00.

16.9.2 Content

Base XSD Type: xsd:hexBinary
length = 4

16.10 kml:coordinatesType

16.10.1 Description

A list of string values representing one or more coordinate tuples, with each tuple consisting of decimal values for geodetic longitude, geodetic latitude, and altitude. The altitude component is optional. The coordinate separator is a comma and the tuple separator is a whitespace. Longitude and latitude coordinates are expressed in decimal degrees only. The radix point (a.k.a 'decimal' mark) is encoded as '.' (i.e. full stop or period symbol).

16.10.2 Content

List of: xsd:string

16.11 kml:dateTimeType

16.11.1 Content

Union of: xsd:dateTime, xsd:date, xsd:gYearMonth, xsd:gYear

16.12 kml:displayModeEnumType

16.12.1 Content

Base XSD Type: xsd:string
value comes from list: {'default'|'hide'}

default Specifies to display the balloon.

hide Specifies to hide the balloon.

16.13 kml:enumBaseType

16.13.1 Description

Serves as a base type for enumeration types.

16.13.2 Content

Base XSD Type: Empty restriction of xsd:string

16.14 kml:flyToModeEnumType

16.14.1 Content

Base Type: kml:enumBaseType
value comes from list: {smooth|bounce}

smooth Specifies a smooth, unbroken flight from through a series of points. An unbroken series of smooth fly-to-modes will not slow to zero at intermediate points.

bounce Specifies a bounce flight mode that begins and ends at zero velocity.

16.15 kml:gridOriginEnumType

16.15.1 Content

Base XSD Type: string
value comes from list: {'lowerLeft'|'upperLeft'}

lowerLeft	Specifies to begin numbering the tiles in a layer of a `kml:ImagePyramid` from the lower left corner.
upperLeft	Specifies to begin numbering the tiles in a layer of `kml:ImagePyramid` from the upper left corner.

16.16 kml:kmlVersionType

16.16.1 Description

Specifies the version of the schema that the KML instance conforms to. A three-level version pattern 'x.y.z' is used, where x=major, y=minor, and z=bug-fix (corrigendum) are integer designations separated by periods. The major and minor version designations shall be provided if the version attribute is specified. The corrigendum designator may be omitted, in which case it shall default to the latest bug-fix version of the schema.

NOTE major.minor is not a decimal number. Version 0.9 is not "almost 1.0" and may be succeeded by version 0.10, 0.11, 0.12 etc on the road to version 1.0.

16.16.2 Content

Base XSD Type: xsd:string
regular expression pattern value: "2.[2-3](.(0|[1-9][0-9]?))?"

16.17 kml:itemIconStateEnumType

16.17.1 Description

Specifies the current state of a `kml:NetworkLink` or `kml:Folder`.

16.17.2 Content

Base XSD Type: xsd:string
value comes from list: {'open'|'closed'|'error'|'fetching0'|'fetching1'|'fetching2'}

open	open folder
closed	closed folder
error	error in fetch

fetching0	fetch state 0
fetching1	fetch state 1
fetching2	fetch state 0

16.18 kml:itemIconStateType

16.18.1 Content

List of: kml:itemIconStateEnumType

16.19 kml:listItemTypeEnumType

16.19.1 Description

Specifies how a *kml:AbstractFeatureGroup* and its contents shall be displayed as items in a list view.

16.19.2 Content

Base XSD Type: string
value comes from list: {'radioFolder'|'check'|'checkHideChildren'|'checkOffOnly'}

radioFolder	Only one of the *kml:AbstractContainerGroup*'s items shall be visible at a time.
check	The *kml:AbstractFeatureGroup's* visibility is tied to its item's checkbox.
checkHideChildren	Use a normal checkbox for visibility but do not display the *kml:AbstractContainerGroup*'s children in the list view. A checkbox allows the user to toggle visibility of the child objects in the viewer.
checkOffOnly	Prevents all items from being made visible at once—that is, the user can turn everything in the *kml:AbstractContainerGroup* off but cannot turn everything on at the same time. This setting is useful for *kml:AbstractContainerGroup*'s containing large amounts of data.

16.20 kml:playModeEnumType

16.20.1 Content

Base Type: kml:enumBaseType
value comes from list: {'pause'}

pause	Specifies to pause the state of play until a user takes action to continue.

16.21 kml:refreshModeEnumType

16.21.1 Content

Base XSD Type: xsd:string
value comes from list: {'onChange'|'onInterval'|'onExpire'}

onChange Refresh when the resource is first loaded and whenever the `kml:Link` parameters change.

onInterval Refresh the resource every *n* seconds as specified in `kml:refreshInterval`.

onExpire Refresh the resource when the expiration time is reached.

16.22 kml:shapeEnumType

16.22.1 Content

Base XSD Type: string
value comes from list: {'rectangle'|'cylinder'|'sphere'}

rectangle Used for an ordinary photo.

cylinder Used for panoramas, which can be either partial or full cylinders.

sphere Used for spherical panoramas.

16.23 kml:styleStateEnumType

16.23.1 Content

Base XSD Type: xsd:string
value comes from list: {'normal'|'highlight'}

normal Specifies a normal style for a kml:Placemark.

highlight Specifies a highlighted style for a kml:Placemark.

16.24 kml:SnippetType

16.24.1 Content

Type: xsd:string

16.24.2 Attributes

16.24.2.1 maxLines

16.24.2.1.1 Description

Specifies the maximum number of lines to display for the *`kml:AbstractFeatureGroup`* `kml:snippet` value in the list view.

16.24.2.1.2 Content

Type:	xsd:int
Default Value:	2

16.25 kml:unitsEnumType

16.25.1 Description

16.25.2 Content

Base XSD Type: xsd:string
value comes from list: {'fraction'|'pixels'|'insetPixels'}

fraction Value is a fraction of the icon.

pixels Value is a specific pixel size.

insetPixels Value is an offset in pixels from the upper right corner of the icon.

16.26 kml:vec2Type

16.26.1 Structure

```
<...
 x="double [0..1]"
 y="double [0..1]"
 xunits=" kml:unitsEnumType [0..1]"
 yunits=" kml:unitsEnumType [0..1]"/>
```

16.26.2 Description

Specifies an image coordinate system.

The *x* and *y* values may each be specified in three different ways: as *pixels* (**pixels**), as fractions of the icon (**fraction**), or as inset pixels (**insetPixels**), which is an offset in pixels from the upper right corner of the icon. They may or may not be specified in a consistent manner - for example, x can be specified in pixels and y as a fraction.

16.26.3 Attributes

16.26.3.1 x

16.26.3.1.1 Description

The *x* component of a point.

16.26.3.1.2 Content

Type:	xsd:double
Default Value:	1.0

16.26.3.2 y

16.26.3.2.1 Description

The *y* component of a point.

16.26.3.2.2 Content

Type:	xsd:double
Default Value:	1.0

16.26.3.3 xunits

16.26.3.3.1 Description

Units in which the *x* value is specified.

16.26.3.3.2 Content

Type:	kml:unitsEnumType
Default Value:	fraction

16.26.3.4 yunits

16.26.3.4.1 Description

Units in which the *y* value is specified.

16.26.3.4.2 Content

FieldType:	kml:unitsEnumType
Default Value:	fraction

16.27 kml:viewRefreshModeEnumType

16.27.1 Content

Base XSD Type: string
value comes from list: {'never'|'onRequest'|'onStop'|'onRegion'}

never	Ignore changes in the geographic view. Also ignore `kml:viewFormat` parameters, if any.
onRequest	Refresh the resource only when the user explicitly requests it.
onStop	Refresh the resource *n* seconds after movement stops, where *n* is specified in `kml:viewRefreshTime`.
onRegion	Refresh the resource if a `kml:Region` becomes active.

17. Media Types

KML documents can be distributed as uncompressed KML or compressed KMZ files.

KML files are plain-text based XML files with a '.kml' file extension. The internet media type: application/vnd.google-earth.kml+xml has been registered with the Internet Assigned Numbers Authority (IANA) for KML files.

KMZ files are ZIP archives with a '.kmz' file extension. The internet media type: application/vnd.google-earth.kmz has been registered with the Internet Assigned Numbers Authority (IANA) for KMZ archives, but application/zip can also be used to identify the entity as a KMZ archive. See Annex C for details regarding the structure and content of a KMZ archive.

Annex A Conformance Class Abstract Test Suite (Normative)

The abstract test suite (ATS) is defined in the companion document OGC 14-068, *OGC KML 2.3 – Abstract Test Suite*. It provides a basis for developing an Executable Test Suite (ETS) that may be used to validate the content and structure of a KML resource.

A KML resource that conforms to this standard shall:

- satisfy all requirements stipulated in this document;
- be well-formed (as defined in the W3C XML 1.0 standard);
- pass all relevant test cases specified by the Abstract Test Suite (ATS) provided in OGC document 14-068.

Annex B KML Coordinate Reference System Definition (Normative)

```
<CompoundCRS gml:id="LonLat84_5773"
  xmlns="http://www.opengis.net/gml/3.2"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  aggregationType="sequence">
  <identifier codeSpace="http://www.opengeospatial.org/ogcna">
  http://www.opengis.net/def/crs/OGC/0/LonLat84_5773
  </identifier>
  <name>Geographic 3D: Long(deg),Lat(deg),Height(m)</name>
  <scope>KML 3D coordinate reference system</scope>
  <componentReferenceSystem>
    <GeodeticCRS gml:id="LonLat84">
      <identifier codeSpace="http://www.opengeospatial.org/ogcna">
      http://www.opengis.net/def/crs/OGC/0/LonLat84
      </identifier>
      <name>WGS 84 with long/lat axis order</name>
      <scope>KML geographic 2D coordinate reference system (adapted
from EPSG-4326).</scope>
      < usesEllipsoidalCS
xlink:href="http://www.opengis.net/def/cs/EPSG/0/6424"
xlink:title=" Ellipsoidal 2D CS. Axes: longitude, latitude.
Orientations: east, north. UoM: degree"/>
      <usesGeodeticDatum xlink:href="
http://www.opengis.net/def/datum/EPSG/0/6326"
        xlink:title="World Geodetic System 1984"
        xlink:role="http://earth-
info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html"/>
    </GeodeticCRS>
  </componentReferenceSystem>
  <componentReferenceSystem>
    <VerticalCRS gml:id="EPSG-5773">
      <identifier codeSpace="http://www.opengeospatial.org/ogcna">
      http://www.opengis.net/def/crs/EPSG/0/5773
      </identifier>
      <name>EGM96 geoid</name>
      <remarks>Height surface resulting from the application of the
EGM96 geoid model to the WGS 84 ellipsoid.</remarks>
      <scope>Geodesy</scope>
      <verticalCS xlink:href="
http://www.opengis.net/def/cs/EPSG/0/6499"
        xlink:title="Gravity-related CS. Axis: height (H). Orientation:
up. UoM: m."/>
      <verticalDatum xlink:href="
http://www.opengis.net/def/datum/EPSG/0/5171"
        xlink:title="EGM96 geoid"
xlink:role="http://cddis.gsfc.nasa.gov/926/egm96/egm96.html"/>
    </VerticalCRS>
  </componentReferenceSystem>
</CompoundCRS>
```

Annex C KMZ Files (Normative)

C.1. **Archive Structure**

The KMZ archive shall contain at least one KML file (the 'main' KML file) and any number of supporting files, such as: overlays, images, icons, models, and other network-linked KML files referenced from the main KML file. The main KML file (a.k.a the root or default KML file) shall be located at the root directory level of the archive and is typically loaded first upon opening the KMZ archive. The conventional name for the main KML file is 'doc.kml', but any local file name can be used. If more than one KML file exists at the root directory level, the first file entry (in unsorted order) shall be treated as the main KML file. To avoid ambiguity, it is strongly recommended that only one KML file should exist at the root directory level.

C.2. **Compression**

Data compression may be used but is not required. At a minimum the ZIP 2.0 compression methods: 'stored' (no compression) and 'deflate' shall be supported. Other ZIP compression methods (e.g. Zip64) may be supported by some implementations but users should check the implementation release notes for compatibility. Note that use of other compression algorithms and/or extensions may not be compatible with all KML viewers.

C.3. **Relative Referencing**

The base URI for resolving relative references is that of the encapsulating KML document entity, i.e. such references are relative to the local file location. The only exception is the `<sourceHref/>` element in the context of a KML model, in which case the base URI of a relative reference is that of the source model file (e.g. .dae file).

It is recommended that special non-alphanumeric characters (e.g. ‘:’, ‘<’. ‘>’, ‘|’) and whitespaces should be avoided in filename entries of the KMZ archive as this can cause issues with resolving references in some implementations.

EXAMPLE: Suppose a KMZ archive contains the following files:

doc.kml
files/doc2.kml
files/image1.png
files/image2.png

The relative URI "files/image1.png" in doc.kml refers to an image resource in the 'files' directory of the archive. In doc2.kml, the relative URI "image2.png" refers to the other image resource in the archive. A relative URI in doc.kml that begins with "../" is a reference whose target is outside of the KMZ archive.

Annex D Extension Model Examples (Informative)

This annex provides examples illustrating the use of the KML Extension Model using extension-by-composition as described in 0.

D.1. **Use of Element Substitution**

In this section, examples of new extension elements are created in a foreign application namespace that substitute for an appropriate abstract KML head element. The examples presented here are inspired by the OGC Change Request 13-033 (Coordinates by reference to other object), which requires further experimentation and implementation in a KML application profile.

Suppose we have `kml:Placemark` representations of two utility/power poles (`P1` and `P2`) and the geometry of each utility pole is defined by a `kml:Point` as shown in the following instance fragment:

```
     <Placemark id="P1">
      <name>Utility Pole #1</name>
      <Point id="Point0001">
       <extrude>true</extrude>
       <altitudeMode>relativeToGround</altitudeMode>
       <coordinates>-123.6526503761486,49.44556103407334,4</coordinates>
      </Point>
     </Placemark>
     <Placemark id="P2">
      <name>Utility Pole #2</name>
      <Point id="Point0002">
       <extrude>true</extrude>
       <altitudeMode>relativeToGround</altitudeMode>
       <coordinates>-123.6527310284922,49.44503199371083,4</coordinates>
      </Point>
     </Placemark>
```

If we want to represent a utility/power line by another `kml:Placemark` that connects the utility poles, `P1` and `P2`, we would normally represent the geometry using a `kml:LineString` and would have to repeat the point coordinate values of `P1` and `P2` in the `kml:coordinates` element as shown in the following instance:

```
  <Placemark id="TL">
   <name>Utility/power Line</name>
   <LineString>
    <coordinates>-123.6526503761486,49.44556103407334,4
     -123.6527310284922,49.44503199371083,4</coordinates>
   </LineString>
  </Placemark>
```

If instead, we wanted to reuse the point positions of `P1` and `P2` without having to repeat the coordinate values in the `kml:LineString`, we could for example create new child extension elements of `kml:LineString` (e.g. representing the start and end points) that can each reference an existing `kml:Point` by its `id` value. To create such an extension element we can make use of the built-in extension points of `kml:LineString`, i.e. substitute for either of the two abstract elements: `kml:LineStringSimpleExtensionGroup` or `kml:LineStringObjectExtensionGroup`. If the new extension element is of complex type (i.e. has its own child element or attributes) then it should substitute for `kml:LineStringObjectExtensionGroup`, otherwise it is of simple type and should substitute for `kml:LineStringSimpleExtensionGroup`. There are several possible approaches, but suppose the desired extension elements are of complex type, called 'StartPoint' and 'EndPoint', and follow the 'reference type' encoding pattern of OGC Geography Markup Language ([8], 7.2.3.7). Then a sample instance of the power line Placemark could be encoded as follows:

```
  <Placemark id="TL">
   <name>Utility/power Line</name>
   <description>A power line that connects two utility poles. The LineString is
specified by extension elements 'ex:StartPoint' and 'ex:EndPoint', which
reference existing kml:Point instances representing utility pole
locations.</description>
   <LineString>
    <ex:StartPoint xlink:href="#Point0001"/>
    <ex:EndPoint xlink:href="#Point0002"/>
   </LineString>
  </Placemark>
```

The extension elements `ex:StartPoint` and `ex:EndPoint` must be declared in an application schema in a foreign namespace (e.g. xmlns:ex="http://www.example.org") and substitute for `kml:LineStringObjectExtensionGroup`. A sample application schema showing one way this can be done is as follows:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.example.org"
 xmlns:kml="http://www.opengis.net/kml/2.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.example.org" elementFormDefault="qualified"
version="0.1.0">

 <import namespace="http://www.opengis.net/kml/2.2"
schemaLocation="ogckml23.xsd"/>
 <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://www.w3.org/1999/xlink.xsd"/>

 <element name="StartPoint" type="ex:ObjectReferenceType"
substitutionGroup="kml:LineStringObjectExtensionGroup"/>

 <element name="EndPoint" type="ex:ObjectReferenceType"
substitutionGroup="kml:LineStringObjectExtensionGroup"/>

 <complexType name="ObjectReferenceType">
  <complexContent>
   <extension base="kml:AbstractObjectType">
    <attributeGroup ref="xlink:simpleAttrs"/>
   </extension>
  </complexContent>
 </complexType>

</schema>
```

Note that the type `ex:ObjectReferenceType` of the new extension elements introduced above, must be validly derived (in this case by extension because attributes are being added) from the type of `kml:LineStringObjectExtensionGroup` (i.e. `kml:AbstractObjectType`). The `ex:ObjectReferenceType` adds the XLink simple attribute group, which includes the `xlink:href` attribute to enable remote referencing.

D.2. Use of Foreign Wildcard Elements

This section illustrates the new extension mechanism introduced in KML 2.3 described in 6.7.2.3 (Elements) as it applies to a proposed KML Application Profile for Enhanced KML Styling Extensions. The proposed Application Profile is work in progress and intends to reuse a subset of schema elements from the existing OGC Symbology Encoding standard to be interleaved with existing KML sub-style elements (e.g. `kml:LineStyle`, `kml:PolyStyle`). This type of extension-by-composition is afforded by the new default open content mechanism in KML 2.3, which was introduced in XML Schema 1.1. The KML 2.3 Schema declares this default open content as shown in the following schema excerpt:

```
  <defaultOpenContent mode="interleave">
    <any namespace="##other" processContents="lax"/>
  </defaultOpenContent>
```

The `<any namespace="##other"/>` part of the declaration allows for any element content from a foreign namespace (i.e. other than the KML namespace). The `processContents="lax"` part of the declaration allows for, but does not require, the foreign content to be validated against the foreign schema and will do so if the location of the schema file is provided in the KML instance document. Setting `mode="interleave"` in the declaration above allows the foreign element content to be placed anywhere in a KML instance document among the existing KML elements.

Reusing XML Content from OGC Symbology Encoding Namespace

The draft Enhance Styling Extension profile proposes to interleave the `se:LineSymbolizer` and `se:PolygonSymbolizer` elements from the OGC Symbology Encoding namespace (xmlns:se="http://www.opengis.net/se") as children of the `kml:LineStyle` and `kml:PolyStyle` elements, respectively. In the case that the `se:LineSymbolizer` element represents a dashed line style, a KML rendering implementation that supports this foreign element may render the corresponding `kml:LineString` as a dashed line. An example of a `se:LineSymbolizer` element representing a simple dashed line style is shown in the following KML instance fragment.

```
<LineStyle>
  <color>8fff0000</color>
  <width>5</width>
  <se:LineSymbolizer xmlns:se="http://www.opengis.net/se">
    <se:Stroke>
      <se:SvgParameter name="stroke">#0000ff</se:SvgParameter>
      <se:SvgParameter name="stroke-width">5</se:SvgParameter>
      <se:SvgParameter name="stroke-opacity">0.5</se:SvgParameter>
      <se:SvgParameter name="stroke-dasharray">5.0 3.0 2.0
3.0</se:SvgParameter>
      <se:SvgParameter name="stroke-linecap">round</se:SvgParameter>
    </se:Stroke>
  </se:LineSymbolizer>
</LineStyle>
```

Similarly, a graphic-filled polygon style might be encoded using a `se:PolygonSybolizer` element encapsulated by a `kml:PolyStyle` as follows:

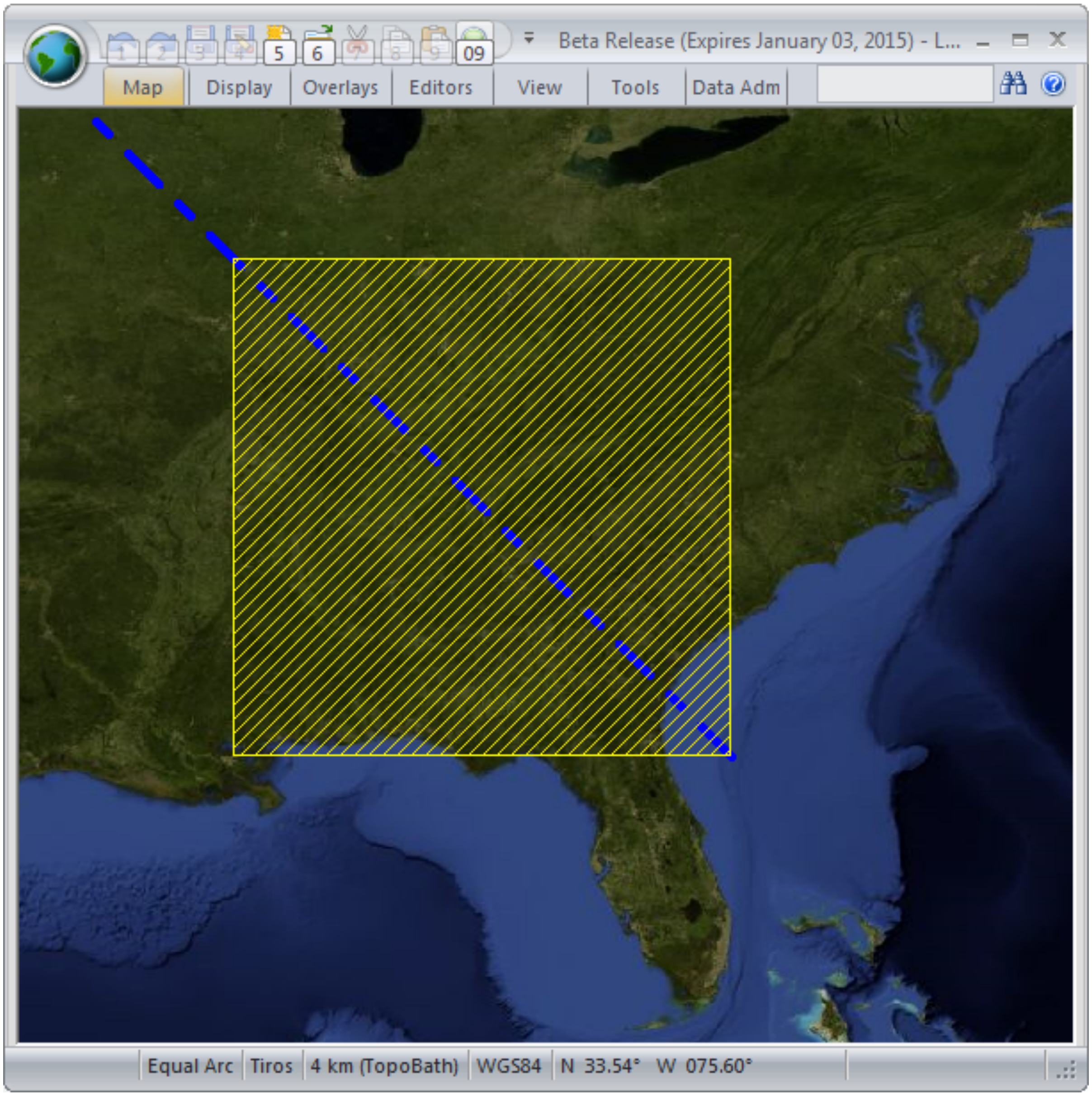
```
<PolyStyle>
  <color>ff0000ff</color>
  <se:PolygonSymbolizer>
    <se:Fill>
      <se:GraphicFill>
        <se:Graphic>
          <se:Mark>
            <se:WellKnownName>triangle</se:WellKnownName>
            <se:Fill>
              <se:SvgParameter name="fill">#ff0000</se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
              <se:SvgParameter name="stroke-width">1</se:SvgParameter>
            </se:Stroke>
          </se:Mark>
          <se:Size>8</se:Size>
        </se:Graphic>
      </se:GraphicFill>
    </se:Fill>
  </se:PolygonSymbolizer>
</PolyStyle>
```

Complete KML Example and Screenshot

A complete KML example that incorporates the `se:LineSymbolizer` and `se:PolygonSymbolizer` extension elements, interleaved as children of the `kml:LineStyle` and `kml:PolyStyle` elements, respectively, is shown in the following instance:

```
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:se="http://www.opengis.net/se">
  <Document>
    <Style id="linestyle">
      <LineStyle>
        <color>8fff0000</color>
        <width>5</width>
        <se:LineSymbolizer>
          <se:Stroke>
            <se:SvgParameter name="stroke">#0000ff</se:SvgParameter>
            <se:SvgParameter name="stroke-width">5</se:SvgParameter>
            <se:SvgParameter name="stroke-opacity">0.5</se:SvgParameter>
            <se:SvgParameter name="stroke-dasharray"> 5.0 3.0 2.0 3.0
</se:SvgParameter>
            <se:SvgParameter name="stroke-linecap">round</se:SvgParameter>
          </se:Stroke>
        </se:LineSymbolizer>
      </LineStyle>
    </Style>
    <Style id="polystyle">
      <LineStyle>
        <color>ff00ffff</color>
        <width>1</width>
      </LineStyle>
      <PolyStyle>
        <color>ff00ffff</color>
        <se:PolygonSymbolizer>
          <se:Fill>
            <se:GraphicFill>
              <se:Graphic>
                <se:ExternalGraphic>
                  <se:InlineContent encoding="base64"
>iVBORw0KGgoAAAANSUhEUgAAAAgAAAAICAYAAADED76LAAAABmJLR0QA/wD/AP+gvaeTAAAACXBIWX
MAAA7DAAAOwwHHb6hkAAAANUlEQVQY033KuREAIAzEQNk5/RdKAUfggOHxKV2FJNpmKB0yFOkQIB2+w
4Xn8ME9NFiDQYAFWrsaAM/GDy8AAAAASUVORK5CYII=</se:InlineContent>
                  <se:Format>image/png</se:Format>
                </se:ExternalGraphic>
                <se:Size>20</se:Size>
              </se:Graphic>
            </se:GraphicFill>
          </se:Fill>
        </se:PolygonSymbolizer>
      </PolyStyle>
    </Style>
    <Style id="polystyle_mark">
      <LineStyle>
        <color>ff000000</color>
        <width>1</width>
      </LineStyle>
      <PolyStyle>
        <color>ff0000ff</color>
        <se:PolygonSymbolizer>
          <se:Fill>
            <se:GraphicFill>
              <se:Graphic>
                <se:Mark>
                  <se:WellKnownName>triangle</se:WellKnownName>
                  <se:Fill>
                    <se:SvgParameter name="fill">#ff0000</se:SvgParameter>
                  </se:Fill>
                  <se:Stroke>
                    <se:SvgParameter name="stroke">#000000</se:SvgParameter>
                    <se:SvgParameter name="stroke-width">1</se:SvgParameter>
                  </se:Stroke>
                </se:Mark>
                <se:Size>8</se:Size>
              </se:Graphic>
            </se:GraphicFill>
          </se:Fill>
        </se:PolygonSymbolizer>
      </PolyStyle>
    </Style>
    <Placemark>
      <styleUrl>#linestyle</styleUrl>
      <LineString>
        <coordinates>-80,30 -100,50</coordinates>
      </LineString>
    </Placemark>
    <Placemark>
      <styleUrl>#polystyle</styleUrl>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates> -90,40,0 -90,30,0 -80,30,0 -80,40,0 -90,40,0
</coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
    <Placemark>
      <styleUrl>#polystyle_mark</styleUrl>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates> -110,40,0 -110,30,0 -100,30,0 -100,40,0 -110,40,0
</coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```

A prototype implementation of FalconView® supports these extensions and renders the KML example instance above as shown in the following screenshot:



D.3. **Use of Foreign Wildcard Attributes**

This section illustrates the new extension mechanism introduced in KML 2.3 described in 6.7.2.4 Foreign Wildcard attributes.

Suppose we wanted to create a KML extension that enables the expression of coordinates of the utility/power pole (`P1`) shown in D.1 using a different Coordinate Reference System (CRS), for examle using the (easting, northing) ordinates of the Universal Transverse Mercator Projection, Zone 10N (with OGC identifier http://www.opengis.net/def/crs/EPSG/0/6339). Assuming that a KML rendering implementation supports such a CRS, a new attribute, say `ex:crs`, could be created on `kml:Point` to encode the CRS identifier value as shown in the following instance:

```
     <Placemark id="P1">
      <name>Utility Pole #1</name>
      <Point id="Point0001" xmlns:ex="http://www.example.org"
ex:crs="http://www.opengis.net/def/crs/EPSG/0/6339">
       <coordinates>452690.92049225536, 5477193.333961494</coordinates>
      </Point>
     </Placemark>
```

Such an extension attribute must be created in a foreign namespace schema (if the attribute can't be reused from an existing third party schema) as shown below:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.example.org"
 xmlns:kml="http://www.opengis.net/kml/2.2"
targetNamespace="http://www.example.org" elementFormDefault="qualified"
version="0.1.0">

 <import namespace="http://www.opengis.net/kml/2.2"
schemaLocation="ogckml23.xsd"/>

 <attribute name="crs" type="anyURI"/>

</schema>
```

Annex E Bibliography

[1] Google Inc., KML Reference Document https://developers.google.com/kml/documentation/kmlreference

[2] IETF RFC 2616, Hypertext Transfer Protocol – HTTP/1.1. (June 1999)

[3] IETF RFC 2806, URLs for Telephone Calls. (April 2000)

[4] ISO 8601:2004, Data elements and interchange formats — Information interchange — Representation of dates and times.

[5] ISO 19101:2002. Geographic information -- Reference model

[6] ISO 19107:2003, Geographic Information — Spatial schema.

[7] ISO 19111:—1), Geographic Information — Spatial referencing by coordinates.

[8] ISO 19136:2007, Geographic information — Geography Markup Language (GML)

[9] OGC 00-014r1, Guidelines for Successful OGC Interface Specifications

[10] OGC 08-125r2, KML Development Best Practices