



GKML

EXPORT DATA TO KML

VOL. 02-52

Ivan V. Dmitriev
06.12.2021

Contents

1. gKml4

2. Functions.....5

 2.1 Open kml5

 2.2 Close kml5

 2.3 Create color5

 2.4 Set Style5

 2.5 Set Point7

 2.6 Set Polyline8

 2.7 Set Polygone.....8

3. Example (gKmlTest)10

Citation.....13

Tables list

Table 1.1 gKml functions.....4

Figures list

Figure 3.1 gKmlTest script result (the 112.kml file is opened in Google Earth) 10

1. gKml

Create kml-file for Google Earth with elementary figures. There are several functions (**Ошибка!** **Источник ссылки не найден.**); each of them sends strings to kml-file.

The function `set` was created using `kml_line` function (<https://www.mathworks.com/matlabcentral/fileexchange/34006-kml-line-plot>).

Table 1.1 gKml functions

Function name	Function description
<code>gKmlClose</code>	Write to kml-file "close lines" and close file
<code>gKmlColor</code>	Create Color&Alpha in kml-format
<code>gKmlOpen</code>	Open file and write to kml-file "open lines"
<code>gKmlSetPoint</code>	Write to kml-file "Point" tags and data
<code>gKmlSetPolygone</code>	Write to kml-file "Polygone" tags and data
<code>gKmlSetPolyline</code>	Write to kml-file "Polyline" tags and data
<code>gKmlSetStyle</code>	Write to kml-file "Style" tags
<code>gKmlTest</code>	Script with gKml example

2. Functions

2.1 Open kml

function fId=gKmlOpen(FileName)

Open file and write to kml-file "open lines".

Parameters:

fId – file identifier;

FileName – the Name of kml-file.

Function Example:

```
>> fId=gKmlOpen('c:\temp\112.kml');
```

2.2 Close kml

function status=gKmlClose(fId)

Write to kml-file "close lines" and close file.

Parameters:

fId – file identifier;

status – returns a status of 0 when the close operation is successful. Otherwise, it returns -1.

Function Example:

```
>> gKmlClose(fId);
```

2.3 Create color

function [LColor,LAlpha]=gKmlColor(LColor,LAlpha)

Create Color&Alpha in kml-format.

Parameters:

LColor – color; the order of expression is bbgrr, where bb=blue (00toff); gg=green (00toff); rr=red (00toff); examples: '00ffff', 'r', [255 255 0];

LAlpha – transparent; for alpha, 00 is fully transparent and ff is fully opaque; ; examples: 'ff', 255.

Function Example:

```
>> [LColor,LAlpha]=gKmlColor('r',255);
```

2.4 Set Style

function gKmlSetStyle(fId,StyleId,varargin)

Write to kml-file "Style" tags; colorMode='normal', gx:labelVisibility=0.

Parameters:

fId – file identifier;

StyleId – the linestyle unique identifier (string or number); examples: 'Stl09', 28, 'RedStyle';

varargin – cells; style parameters for elements 'LineStyle', 'BalloonStyle'; example: {'LineStyle',4,'r','ff'}

if parameter is empty, then it is not set to kml-file (Color and Alpha must be empty both);

'LineStyle' >>>> Specifies the drawing style (color, color mode, and line width) for all line geometry. Line geometry includes the outlines of outlined polygons and the extruded "tether" of Placemark icons (if extrusion is enabled).

LWidth – width of the line, in pixels; examples: '05', 4;

LColor – line color; the order of expression is bbgrr, where bb=blue(00toff); gg=green(00toff); rr=red(00toff); examples: '00ffff', 'r', [255 255 0];

LAlpha – line transparent; for alpha, '00' is fully transparent and 'ff' is fully opaque; examples: 'ff', 255.

'PolyStyle' >>>> Specifies the drawing style for all polygons, including polygon extrusions (which look like the walls of buildings) and line extrusions (which look like solid fences).

PColor – polygon color;

PAlpha – polygon transparent;

PFill – Boolean value. Specifies whether to fill the polygon; example: '1',0;

POutline – Boolean value. Specifies whether to outline the polygon. Polygon outlines use the current LineStyle; example: '1',0;

'BalloonStyle' >>>> Specifies how the description balloon for placemarks is drawn. The <bgColor>, if specified, is used as the background color of the balloon.

bgColor – background color; the order of expression is bbgrr, where bb=blue (00toff); gg=green (00toff); rr=red (00toff); examples: '00ffff', 'r', [255 255 0];

bgAlpha – background transparent; for alpha, '00' is fully transparent and 'ff' is fully opaque; examples: 'ff', 255;

textColor – text color;

textAlpha – text transparent;

text – Text displayed in the balloon; see <https://developers.google.com/kml/documentation/kmlreference#balloonstyle> for details.

text example – <![CDATA[\${name}
 \${description}]]>

'IconStyle' >>>> Specifies how icons for point Placemarks are drawn, both in the Places panel and in the 3D viewer of Google Earth. The <Icon> element specifies the icon image. The <scale> element specifies the x, y scaling of the icon. The color specified in the <color> element of <IconStyle> is blended with the color of the <Icon>.

iColor – icon color;

iAlpha – icon transparent;

iScale – resizes the icon (default=1);

iHeading – compass direction, in degrees; default=0 (North); values range from 0 to +-180 degrees;

iHref – an HTTP address or a local file specification used to load an icon; example:

`http://maps.google.com/mapfiles/kml/pal3/icon21.png`

hS – specifies the position within the Icon that is "anchored" to the <Point> specified in the Placemark

hS={hS_x,hS_y,hS_xunits,hS_yunits}; Specifies the position within the Icon that is "anchored" to the <Point> specified in the Placemark. The x and y values can 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. The x and y positions can be specified in different ways—for example, x can be in pixels and y can be a fraction. The origin of the coordinate system is in the lower left corner of the icon.

x – Either the number of pixels, a fractional component of the icon, or a pixel inset indicating the x component of a point on the icon.

y – Either the number of pixels, a fractional component of the icon, or a pixel inset indicating the y component of a point on the icon.

xunits – Units in which the x value is specified. A value of fraction indicates the x value is a fraction of the icon. A value of pixels indicates the x value in pixels. A value of insetPixels indicates the indent from the right edge of the icon.

yunits – Units in which the y value is specified. A value of fraction indicates the y value is a fraction of the icon. A value of pixels indicates the y value in pixels. A value of insetPixels indicates the indent from the top edge of the icon.

'LabelStyle' >>>> Specifies how the <name> of a Feature is drawn in the 3D viewer. A custom color, color mode, and scale for the label (name) can be specified.

LbColor – label color;

LbAlpha – label transparent;

LbScale – resizes the label.

Function Example:

```
>> gKmlSetStyle(fId,'st001',{ 'LineStyle',4,'r','ff'},{ 'BalloonStyle','c','0f','b','ff','Area01'});
```

```
>> gKmlSetStyle(fId,'style02',{ 'LineStyle',4,'r','ff'},{ 'PolyStyle','b','5f','1','0'});
```

2.5 Set Point

function gKmlSetPoint(fId,B,L,Z,altitudeMode,PlaceName,StyleId,Descript)

Write to kml-file "Point" tags and data; extrude=0.

Parameters:

fId – file identifier;

B – latitude in degree, one number;

L – longitude in degree; one number;

Z – coordinate along Z-axis depend on altitudeMode; if empty, than ignored (clampToGround and clampToSeaFloor mode used);

altitudeMode – specifies how altitude components in the <coordinates> element are interpreted; there are

(1)clampToGround and clampToSeaFloor, (2)relativeToGround and relativeToSeaFloor, (3)absolute

PlaceName – the name of place; examples: 'Line002', 28;

PointId – the name of point; examples: 'Line002', 28;

LineStyleId – the linestyle unic identifier (string or number); examples: 'Stl09', 28, 'RedStyle';

Descript – description in a separate window; examples: 'seismic survey, area B005',

'<![CDATA[Multichannel Seismic data

]]>'

Function Example:

```
>> gKmlSetPoint(fId,65.1,141,10,2,'Line023','B1','1234567890');
```

2.6 Set Polyline

function gKmlSetPolyline(fId,B,L,Z,altitudeMode,PlaceName,LineStyleId,Descript)

Write to kml-file "Polyline" tags and data; extrude=0, tessellate=1.

Parameters:

fId – file identifier;

B – latitude vector in degree;

L – longitude vector in degree;

Z – coordinate vector along Z-axis depend on altitudeMode; if empty, than ignored (clampToGround and clampToSeaFloor mode used);

altitudeMode – specifies how altitude components in the <coordinates> element are interpreted; there are

(1)clampToGround and clampToSeaFloor, (2)relativeToGround and relativeToSeaFloor, (3)absolute

PlaceName – the name of place (line); examples: 'Line002', 28;

LineStyleId – the linestyle unic identifier (string or number); examples: 'Stl09', 28, 'RedStyle';

Descript – description in a separate window; examples: 'seismic survey, area B005',

'<![CDATA[Multichannel Seismic data

]]>'

Function Example:

```
>> gKmlSetPolyline(fId,[65.1 65.2 65.3],[141 141 142],[],1,'Line023','st001','');
```

2.7 Set Polygone

function gKmlSetPolygone(fId,B,L,Z,altitudeMode,PlaceName,LineStyleId,Descript,varargin)

Write to kml-file "Polygone" tags and data; extrude=0, tessellate=1. A Polygon is defined by an outer boundary and 0 or more inner boundaries; the <coordinates> for polygons must be specified in counterclockwise order; the last point must be equal first point

A Polygon is defined by an outer boundary and 0 or more inner boundaries. The boundaries, in turn, are defined by LinearRings.

When a Polygon is extruded, its boundaries are connected to the ground to form additional polygons, which gives the appearance of a building or a box. Extruded Polygons use <PolyStyle> for their color, color mode, and fill.

The <coordinates> for polygons must be specified in counterclockwise order. Polygons follow the "right-hand rule," which states that if you place the fingers of your right hand in the direction in which the coordinates are specified, your thumb points in the general direction of the geometric normal for the polygon. (In 3D graphics, the geometric normal is used for lighting and points away from the front face of the polygon.)

Since Google Earth fills only the front face of polygons, you will achieve the desired effect only when the coordinates are specified in the proper order. Otherwise, the polygon will be gray.

Parameters:

fId – file identifier;

B – latitude vector in degree;

L – longitude vector in degree;

Z – coordinate vector along Z-axis depend on altitudeMode; if empty, than ignored (clampToGround and clampToSeaFloor mode used);

altitudeMode – specifies how altitude components in the <coordinates> element are interpreted; there are

(1)clampToGround and clampToSeaFloor, (2)relativeToGround and relativeToSeaFloor, (3)absolute

PlaceName – the name of place (Polygone); examples: 'Polygone002', 28;

LineStyleId – the linestyle unic identificator (string or number); examples: 'Stl09', 28, 'RedStyle';

Descript – description in a separate window; examples: 'seismic survey, area B005',
'<![CDATA[Multichannel Seismic data
]]>'

varargin – a number of cells with "innerBoundary"; cells format is {B,L,Z} or {B,L,[]};

Function Example:

```
>> gKmlSetPolygone(fId,[64 64.5 63.5 64],[143 140 139 143],[],1,'Polygone01','style02','Polygone  
example with style02',{[64 64 63.8],[140.8 140.25 140.3],[]},{[63.98 63.62 63.78],[140 139.30  
140.19],[]});
```

3. Example (gKmlTest)

Test functions from gKml functions set.

```
>> fId=gKmlOpen('c:\temp\112.kml');
>> gKmlSetStyle(fId,'style01',{'LineStyle',4,'r','ff',{'BalloonStyle','c','0f','b','0f','<![CDATA[<b><font
color="#CC0000" size="-3">${name}</b> <br> ${description}]]>'});
>> gKmlSetStyle(fId,'style02',{'LineStyle',4,'r','ff',{'PolyStyle','b','5f','1','0'});
>> gKmlSetPoint(fId,64.85,140.34,10,2,'Line023','style01','Point example with style01');
>> gKmlSetPolyline(fId,[65.1 65.2 65.5],[141 140.5 142],[,],1,'Line023','style01','Line examle with
style01');
>> gKmlSetPolygone(fId,[64 64.5 63.5 64],[143 140 139 143],[,],1,'Polygone01','style02','Polygone
examle with style02',{[64 64 63.8],[140.8 140.25 140.3],[,]},{[63.98 63.62 63.78],[140 139.30
140.19],[,]});
gKmlClose(fId);
```

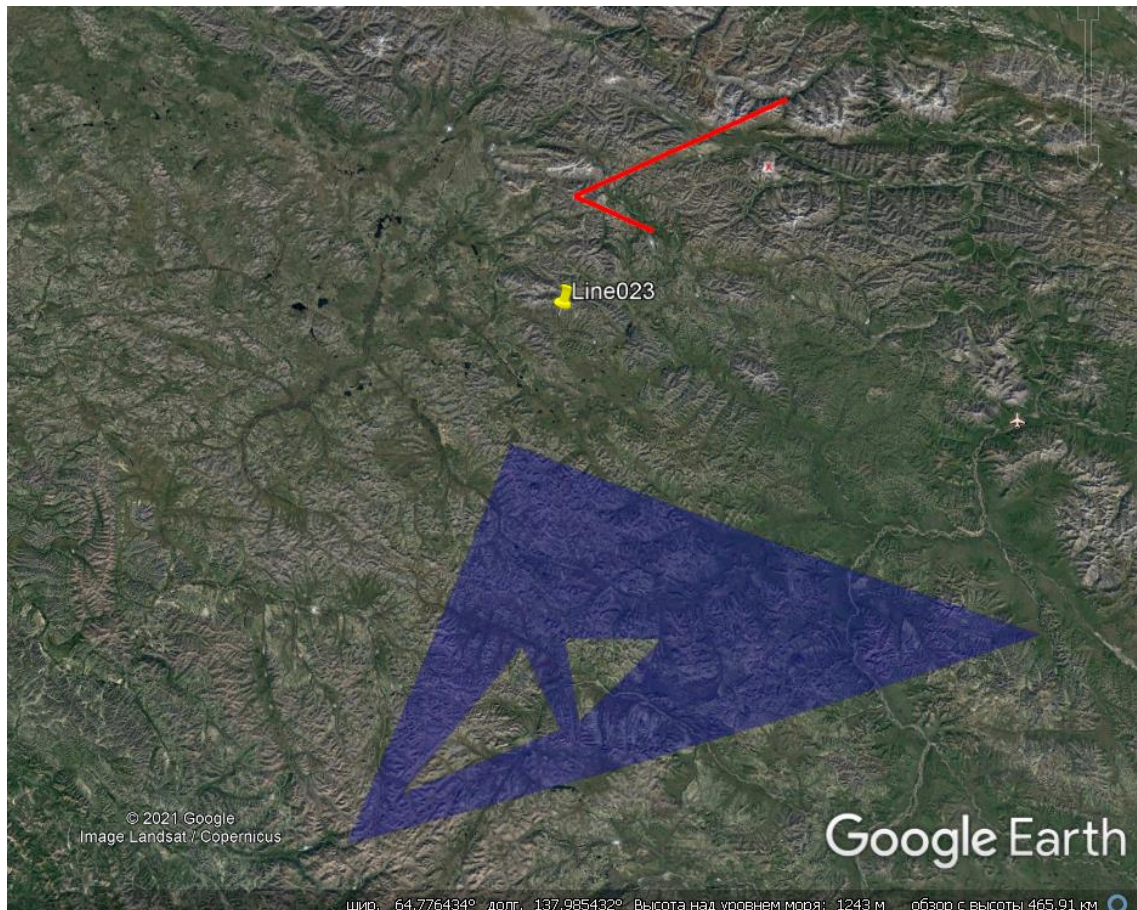


Figure 3.1 gKmlTest script result (the 112.kml file is opened in Google Earth)

The 112.kml file's text was created by gKmlTest script.

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2" xmlns:kml="http://www.opengis.net/kml/2.2"
xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
  <name>112.kml</name>
  <Style id="style01">
    <LineStyle>
      <colorMode>normal</colorMode>
      <color>#ff0000ff</color>
      <width>4</width>
      <gx:labelVisibility>0</gx:labelVisibility>
    </LineStyle>
    <BalloonStyle>
      <bgColor>#0ffff0</bgColor>
      <textColor>#0fff0000</textColor>
      <text><![CDATA[<b>font color="#CC0000" size="-3">${name}</b> <br>
${description}]]></text>
      <displayMode>default</displayMode>
    </BalloonStyle>
  </Style>
  <Style id="style02">
    <LineStyle>
      <colorMode>normal</colorMode>
      <color>#ff0000ff</color>
      <width>4</width>
      <gx:labelVisibility>0</gx:labelVisibility>
    </LineStyle>
    <PolyStyle>
      <colorMode>normal</colorMode>
      <color>#5fff0000</color>
      <fill>1</fill>
      <outline>0</outline>
    </PolyStyle>
  </Style>
  <Placemark>
    <name>Line023</name>
    <description>Point example with style01</description>
    <styleUrl>#style01</styleUrl>
    <Point>
      <extrude>0</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>
      <coordinates>
        140.34,64.85,10
      </coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>Line023</name>
    <description>Line examle with style01</description>
    <styleUrl>#style01</styleUrl>
    <LineString>
      <extrude>0</extrude>
      <tessellate>1</tessellate>
      <altitudeMode>clampToGround</altitudeMode>
      <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>
      <coordinates>
        141,65.1,0,140.5,65.2,0,142,65.5,0
      </coordinates>
    </LineString>
  </Placemark>
  <Placemark>
    <name>Polygone01</name>
    <description>Polygone examle with style02</description>
    <styleUrl>#style02</styleUrl>
    <Polygon>
      <extrude>0</extrude>
      <tessellate>1</tessellate>
      <altitudeMode>clampToGround</altitudeMode>
      <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <altitudeMode>clampToGround</altitudeMode>
          <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>
          <coordinates>
            143,64,0,140,64.5,0,139,63.5,0,143,64,0
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark>

```

```

        </LinearRing>
    </outerBoundaryIs>
    <innerBoundaryIs>
        <LinearRing>
            <altitudeMode>clampToGround</altitudeMode>
            <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>
            <coordinates>
                140.8,64,0,140.25,64,0,140.3,63.8,0
            </coordinates>
        </LinearRing>
    </innerBoundaryIs>
    <innerBoundaryIs>
        <LinearRing>
            <altitudeMode>clampToGround</altitudeMode>
            <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>
            <coordinates>
                140,63.98,0,139.3,63.62,0,140.19,63.78,0
            </coordinates>
        </LinearRing>
    </innerBoundaryIs>
</Polygon>
</Placemark>
</Document>
</kml>

```

Citation

- 1) <https://developers.google.com/kml/documentation/kmlreference>