

UNIVERSITY OF MINNESOTA

# KML TOOLKIT <sup>1</sup>

*VikrantKrishna, kris0176@cs.umn.edu*

*VasudhaMithal, vasudham@cs.iitk.ac.in*

University of Minnesota, Department of Computer Science  
Minneapolis, MN 55455  
July, 2009

---

<sup>1</sup>Kml Toolkit is copyrighted by the regents of the University of Minnesota.

# 1. INTRODUCTION

KML or Keyhole Markup Language is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. It is an XML-based language and uses a tag-based structure. Just as web browsers render HTML, Earth browsers render KML.

## 1.1 About Kml Toolkit

Kml Toolkit is a package to generate kmz files (a zipped format of kml files). It extracts information from text files and generates a kmz file which can be rendered in Earth browsers.

## 1.2 Using the manual

This manual is categorized as follows: *section 2* contains compilation and installation instructions; *section 3* contains the usage information. It has details of the input and output file formats and contains information about the usage of the output file; *section 4* contains the contacts to report any bugs or for any query; finally *section 5* contains copyright information and references.

Symbollic words are enclosed within underscores. For eg. `_filename_` refers to some file name.

# 2. INSTALLATION & COMPILATION

## 2.1 Installation

The package consists of five python modules and a utilities folder. Extract these and the input files in the same directory:

- Event2Kml.py - main python module to generate the kmz file.
- GenerateInput.py - secondary python module to acquire user input and check its validity.
- FileReader.py - secondary python module to read input file and extract data.
- Assets.py - secondary python module with added utilities.
- Kml\_constants.py - secondary python module to write parts of kml code.
- Resources - utilities folder containing required images and font files.

The package uses python libraries like sys, os, optparse, datetime, time, getopt, xml.dom, tempfile, zipfile, shutil, Image, ImageDraw & ImageFont. Make sure these are installed in the system.

## 2.2 Compilation

After installation, run the following commands:

*cd < path >*

*pythonEvent2Kml.py < options >*

- *\_path\_* refers to the directory path where the package has been installed.
- *\_options\_* include: --datafile, --eventfile, --polygon, -- noanimation, --legendoff, --outputfile, --url, --marksize, --help
  - At least one input in the form of datafile, eventfile or polygon is required.
  - use --datafile=*\_filename\_* to supply datafiles.
  - use --eventfile=*\_filename\_* to supply eventfiles.
  - use --polygon=*\_filename\_* to supply polygons.
  - use --noanimation to display events without animation.
  - use --legendoff to display events without color description box.
  - use --outputfile=*\_filename\_* to set name of output file.
  - use --url=*\_url\_* to give url of time series curves.
  - use --marksize=*\_marker\_size\_* to set marker size for datafiles.
  - use --help for further information.

For example

```
*****
python Event2Kml.py --datafile=file1.txt --datafile=file2.txt --polygon=file3.txt
--outputfile=fires --url=http://gofc.cs.umn.edu/EVI_PLOTTER/plot.php
*****
```

provides file1.txt, file2.txt and file3.txt as inputs to the module Event2Kml.py and produces fires.kmz which can be rendered in earth browsers. The url contains time series curves which can be seen by clicking the markers in earth browsers.

Make sure that the color sequence written in the colormap dictionary in Event2Kml.py is the same as the desired coloring sequence of input datafiles. The module Event2Kml.py can be opened with any text editor.

```

*****
python Event2Kml.py --eventfile=file1.txt --outputfile=animated_fires --
url=http://gofc.cs.umn.edu/EVI_PLOTTER/plot.php
*****
provides file1.txt as input to Event2Kml.py and generates animated out-
put file animated_fires.kmz. The url contains time series curves which
can be seen by clicking the markers in earth browsers.

*****
python Event2Kml.py --eventfile=file1.txt --noanimation --outputfile=static_fires
--url=http://gofc.cs.umn.edu/EVI_PLOTTER/plot.php
*****
provides file1.txt as input to Event2Kml.py and generates static output
file static_fires.kmz. The url contains time series curves which can be seen
by clicking the markers in earth browsers.

```

### 3. USING KML TOOLKIT

This section describes formats of various input files valid for Kml Toolkit and the usage of output file.

#### 3.1 Input Files

##### 3.1.1 Event Files

These are the text files which can produce animated or static (as specified by user) kmz files.

They contain entries of the form:

*$\langle latitude, longitude, date, burn - value \rangle$*

- *latitude, longitude* are geographical coordinates of the place where the event had occurred.
- *date* is the date (in dd-mm-yy format) when the event occurred. This helps to create labels in Earth browser as the timeline progresses.
- *Burn-value* is the score generated by data-mining algorithms which is an index of the extent of fire at the location. This is used to color the icons according to thresholds defined in 'legendmap' dictionary in the Event2Kml.py module.

In addition to these entries, a line (to be displayed as the folder name in places panel in Earth browser) can be added at the top.

*Note :* The line should start after '#' followed by a blank space.

*Sample eventfile:*

```
# Fires detected using yearly delta algorithm
35.032292,-119.068715,07-Mar-2003,4679.190476
35.032292,-119.071259,23-Mar-2003,4522.761905
35.030208,-119.070768,19-Feb-2003,4518.190476
```

One can provide multiple eventfiles. `_name_` sets different icons for different files. The image files for icons are present inside the Resources folder.

- First file: the 'Fire' icon (Fire.png)
- Second file: the 'Tree' icon (Tree.png)
- Third file: the 'Flag' icon (Flag.png)
- Fourth file: the 'Plus' icon (Plus.png)
- Fifth file: the 'Star' icon (Star.png)

Color of the icon is set according to the burn-value of the event. Four different colors with rgb codes as follows are possible: fff2b, efccc5, ee0400, 776221. Burn-value thresholds for coloring can be defined in the 'legendmap' dictionary in Event2Kml.py module.

### 3.1.2 Data Files

These are the text files which can produce static kmz files. They contain entries of the form:

$\langle latitude, longitude \rangle$

- *latitude, longitude* are geographical coordinates of the place where the event had occurred.

In addition to these entries, a line (to be displayed as the folder name in places panel in Earth browser) can be added as a commented line at the top.

*Note :* The line should start after '#' followed by a blank space.

*Sample datafile:*

```
# Fires detected using recursive merging algorithm
35.032292,-119.142499,
35.030208,-119.139462,
35.032452,-119.149342,
35.034565,-119.137235,
```

One can have multiple datafiles. Kml Toolkit uses the icon datafileMarker.png to mark the locations. The color of the icon is set according to the sequence in which datafiles are supplied. The datafiles are mapped to respective colors according to the 'colormap' dictionary.

For example if the user runs the command:

\*\*\*\*\*

```
python Event2Kml.py --datafile=file1.txt --datafile=file2.txt
--outputfile=Fire
```

\*\*\*\*\*

Kml Toolkit will set the color of file1.txt icon as specified in colormap['1'] and color of file2.txt icon as specified in colormap['2'] and generate the output file Fire.kmz

User can edit the dictionary in case the sequence written in it is not the desired one. The get\_colorcode() function in Assets.py module can be extended to add more colors. The modules are well documented. Please refer to associated docstrings for further information.

Detailed description of color code used in 'colormap' dictionary:

Code	Color-name	Hex-value	Color
dr	dark-red	660000	
mr	mid-red	cc0000	
lr	light-red	ff6666	
dg	dark-green	006600	
mg	mid-green	33cc00	
lg	light-green	ccff66	
dy	dark-yellow	ff9900	
my	mid-yellow	ffcc33	
ly	light-yellow	ffff99	
db	dark-blue	000099	
mb	mid-blue	0000ff	
lb	light-blue	33ffff	
dp	dark-pink	990066	
mp	mid-pink	ff33ff	
lp	light-pink	ff99ff	
w	white	ffffff	
purple	purple	660099	
olive	olive	808000	

### 3.1.3 Polygons

These are the kml files which mark polygons on earth browsers.

## 3.2 Output File

Kml Toolkit generates files in kmz format which is a zipped form of kml files. The filename is set according to the --outputfile option. In case --outputfile option is not used by the user, Kml Toolkit sets it to the default filename document.kmz

The output files can be rendered by earth browsers. The eventfiles display a box titled 'Legend' which displays information about icon - *burn value* mapping. The animated eventfiles display date below the box as the timeline progresses.

For further information on kml, refer to the following link:

[http://code.google.com/apis/kml/documentation/kml\\_tut.html](http://code.google.com/apis/kml/documentation/kml_tut.html)

## 4. CONTACTS

Kml Toolkit has no known bugs but in case of any problem or query, please send a mail to vasudham AT cs DOT iitk DOT ac DOT in. Refer to <http://vk.cs.umn.edu/KML-TOOLKIT/index.htm> for further information.

## 5. COPYRIGHT & REFERENCES

### 5.1 Copyright

Copyright (C) 2009, the University of Minnesota

Kml Toolkit is a free package; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation [version 2 of the License, or any later version]

For details, browse GNU Public License site

For additional information, contact the Free Software Foundation Inc., 65 Mass Ave, Cambridge, MA 02139, USA.

## 5.2 References

1. For kml: [http://code.google.com/apis/kml/documentation/kml\\_tut.html](http://code.google.com/apis/kml/documentation/kml_tut.html)
2. For Python: [www.python.org](http://www.python.org)