Using Google Earth to Access and Display Emissions Data

David Mintz
U.S. Environmental Protection Agency
Office of Air Quality Planning and Standards
Research Triangle Park, NC 27711
mintz.david@epa.gov

ABSTRACT

If you have used Google Earth™ to look at your neighbor's backyard or fly though the Grand Canyon, then you're ready for this paper. The EPA's Emissions Inventory and Analysis Group is building linkages to emissions data. If you can navigate Google Earth, you will soon be able to access and display emissions data quite easily. This paper provides some background regarding Google Earth's Keyhole Markup Language (KML). It also describes how SAS® is used to access and display data within the Google Earth application.

INTRODUCTION

In the world of data analysis there are many powerful tools. SAS is one of those tools. SAS/IntrNet® software, in particular, allows users to get results based on selected parameters. In the world of spatial analysis, there are many powerful tools. Google Earth is one of those tools. This paper explores how merging SAS and Google Earth results in a very powerful, yet very simple, application.

BODY

Setting the Stage

Before discussing the "nuts and bolts" of how this application works, I'd like to emphasize that the end user needs only one software package – a licensed copy of Google Earth. The data connectivity happens "behind the scenes" and is completely transparent to the end user. It is part of the simplicity of this application. For it to work, however, a developer must do a little work in advance to set up this process. Specifically, the following components are required:

- A KML file that points to a SAS/IntrNet service
- A working SAS/IntrNet service with connectivity to the source data
- A SAS program that can access the source data and generate the desired output

Once these components are in place, anyone can use the KML file to explore the locations and access data dynamically (i.e. when the request is made). Again, the fact that it is connected to a SAS service is completely transparent. The user's copy of Google Earth will open the KML file. Figure 1 provides a complete flow chart.

What is KML?

Google Earth reads files that are written in KML much like a web browser reads files written in HTML. KML is a mark-up language that uses tags to describe elements. You basically have a "placemark" tag for each point you want to plot. Among other things, these tags allow you to specify where the placemark icons are plotted. The tags allow you to control shapes, sizes, and colors of the placemark icons. And perhaps most important and relevant to the topic of this paper, they allow you to place images in the placemark's description balloon (see Figure 2 for a terminology diagram).

Since you can link to a *static* image inside the description balloon, there's no reason you can't link to a *dynamic* image inside the description balloon. Herein lies the key to linking Google Earth with SAS. This is explained further in the "Linking to SAS and source data" section.

Building a KML File

So how do you build the KML file? There are software packages that can help put your data into KML. Another way is to start from scratch and code the file yourself, one line at a time. This approach is fine if you have a lot of time on your hands, which most of us, myself included, do not. Since I'm a programmer, I make SAS do all the hard work. I have a program that writes the KML header and footer and all the stuff in between. The in-between stuff is where the SAS code comes in handy because I have SAS loop through all my points (rows of data) and output a placemark tag for each one. The neat thing about doing it this way is that you can conditionally control the attributes of the placemark. That is, you can change the placemark icon's shape, size, or color based on the data. You can also scale the height of a placemark based on the data. Figure 3 provides some examples. Here, I've plotted electric generating units (EGUs), more generically called power plants, as yellow icons and non-EGUs as red icons. I've also scaled the height to be proportional to the total tons of emissions at the facility in 2002. Right away, you can tell where the highest emitters are and whether they are EGUs. This is a rather non-traditional way of making use of Google Earth's 3-D viewing capability.

Linking to SAS and Source Data

The key to linking to SAS and ultimately the source data is one simple piece of code that goes in the placemark tag in the KML file. For this to work you must already have a SAS/IntrNet service available (that's a topic for another paper). This piece of code is inserted into the image tag. The code, instead of pointing to a static jpg or gif file, points to a SAS/IntrNet service. The code also identifies the SAS program and the name-value pairs to be passed to the SAS program. The looping code that was discussed earlier puts a distinct identifier (facility code in this example) in each placemark tag. Now, when the end user clicks a particular placemark icon in GE, the distinct facility code is passed to the SAS program.

What Happens in the SAS Program

What actually happens when the placemark icon is clicked and the information is passed to the SAS program? The program must be set up in advance to accept the name-value pairs. The program is also where the data access takes place. Again, this requires SAS/IntrNet and SAS/Access software being installed and configured on the Web server or another server accessible to the Web server. There is an SQL procedure in the SAS code that points to the database and queries the appropriate data. For efficiency, only the data for the selected facility is queried. So the facility id is placed in the 'WHERE' clause. Once the data are returned, the program uses the data to generate a plot or table or whatever the developer programmed to be output to the placemark's description balloon. Note that the output is generated dynamically. There are many advantages of accessing the data dynamically. For example, if you have data that are updated frequently, dynamic access will ensure that your output is based on the most recent data. Another reason good reason for dynamic access is that, as a developer, you might need to change the format of your output. Instead of having to change hundreds or thousands of graphs, you only have to change your code. Once the code change is made, future calls to the program will display all graphs with your change. This efficiency, in my opinion, is the most compelling reason to use dynamic access.

Displaying Results in the Description Balloon

Once the graphic is generated, it is displayed in the description balloon. How long it takes for the graphic to appear in the description balloon depends on several factors (e.g. amount of data queried, number and type of data processes, server performance). If you are accessing a reasonably small amount of data, the graphic should appear within seconds and in many cases almost instantaneously when the placemark icon is clicked.

Figure 4 shows the type of plot that is generated and displayed when one of the placemark icons is clicked. The plot provides the name of the facility and how many tons of SO2 and NOX (and other pollutants) were emitted in 2002. This is just an example of the type of output that can be provided. Since the output is coded in SAS, the developer of the SAS code can control how it is displayed.

CONCLUSIONS

This application merges two powerful software products and leverages their strengths to create a super-powerful, yet super-simple analysis tool. Because Google Earth is so intuitive, many data providers are beginning to offer their data in KML. This paper provides one example of an application for air quality analysis, but it just scratches the surface. There is great potential for applications like these to be utilized increasingly in the future.

REFERENCES

Google Earth KML documentation on World Wide Web, http://code.google.com/apis/kml/documentation/.

SAS Web Tools: Advanced Dynamic Solutions Using SAS/IntrNet Software Course Notes, SAS Institute Inc., Cary, NC, 2001.

ACKNOWLEDGMENTS

I would like to thank Scott Goodrick (USDA Forest Service) and Joe Abraham (U.S. EPA, Region 9) for helping me begin to understand the inner workings of Google Earth, Michael Rizzo (U.S. EPA, Office of Air Quality Planning and Standards) for insights regarding KML file-building using SAS software, and Nick Mangus (U.S. EPA, Office of Air Quality Planning and Standards) for actually coming up with the idea of placing SAS output inside the description balloon. I would also like to thank Michelle Wayland for reviewing and commenting on this paper, and the members of my group at EPA for ideas about how to display the data.

DISCLAIMER

This paper represents the views of the author and not necessarily those of U.S. Environmental Protection Agency (EPA). Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

KEY WORDS

Google Earth
SAS
GIS
Emissions
Air Quality
Data Analysis
Mapping
Spatial Analysis

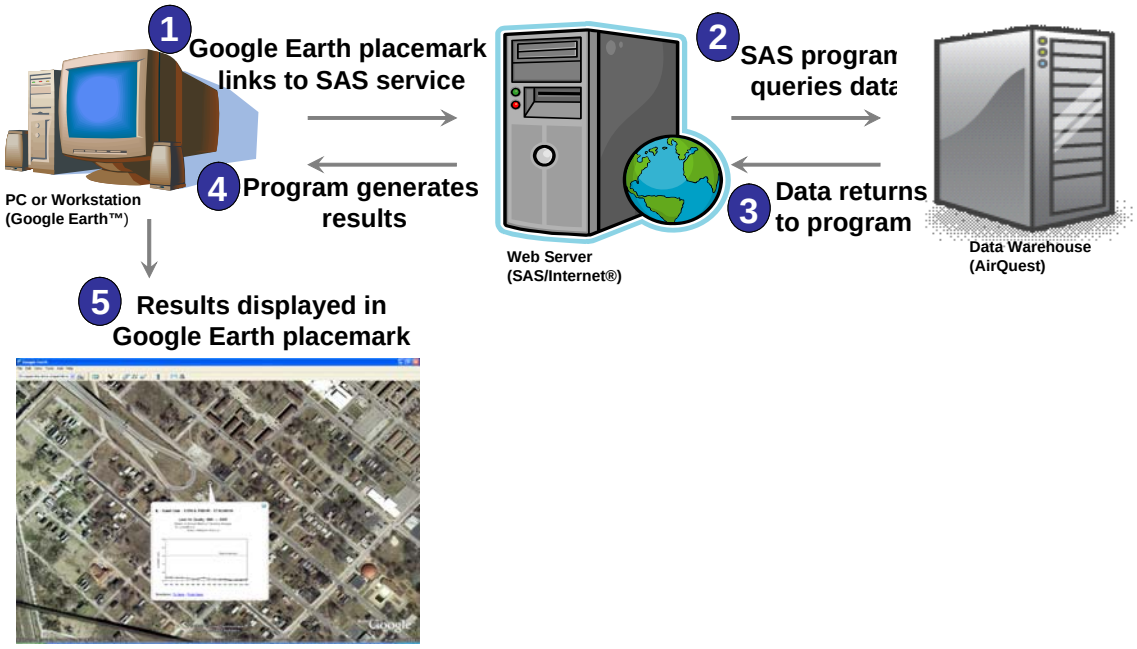

Figure 1. Flow chart explaining how Google Earth can be linked to SAS/InterNet service.

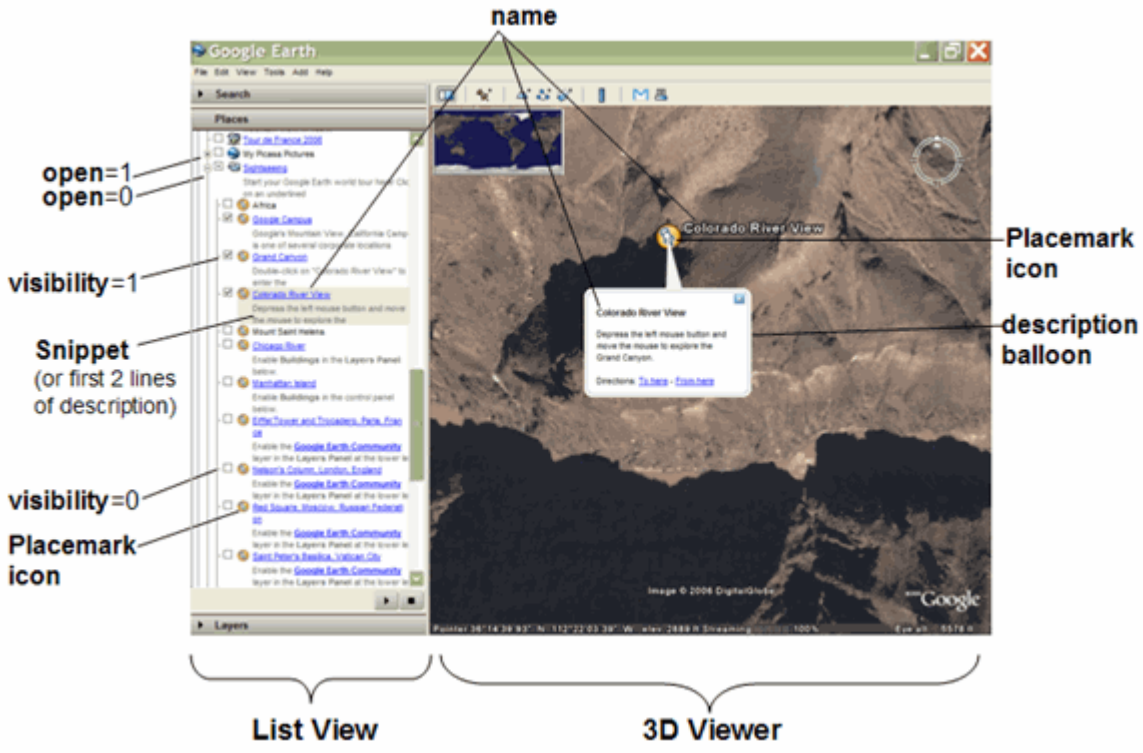

Figure 2. Google Earth terminology and reference diagram. Source: http://code.google.com/apis/kml/documentation/kml_tags_21.html

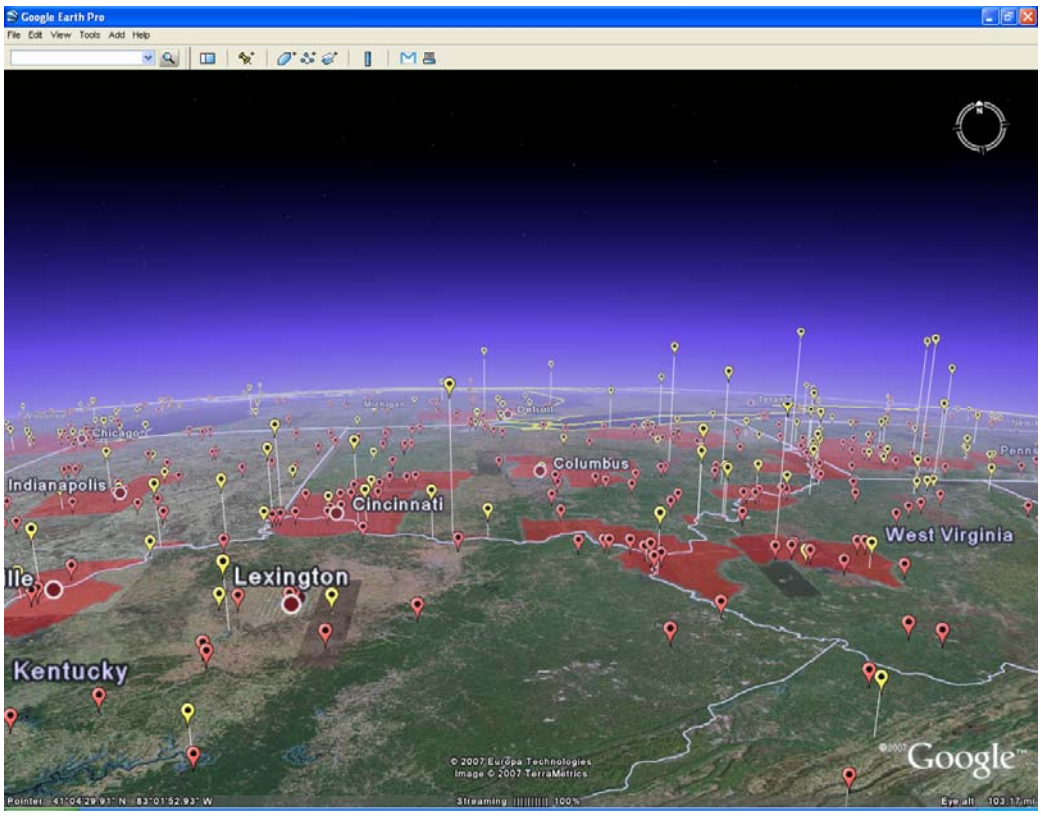

Figure 3. Google Earth displaying KML of emissions point sources.

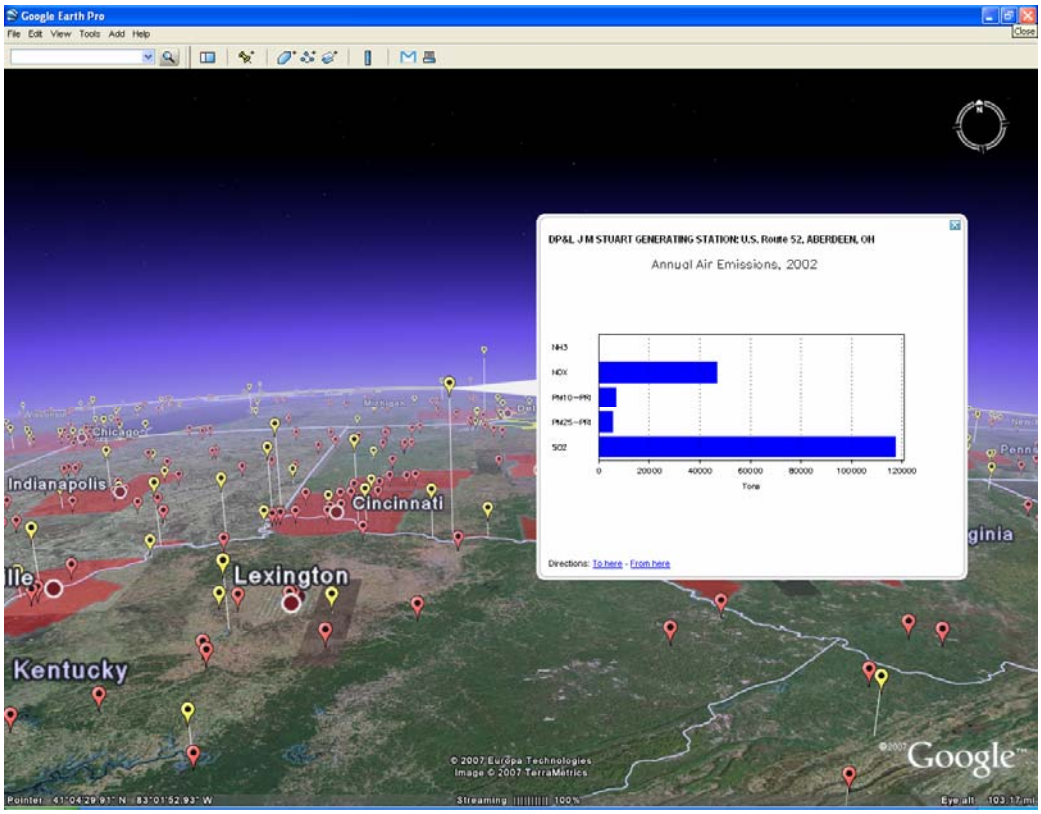

Figure 4. Google Earth displaying KML of emissions point sources, including a description balloon with SAS output.