



## Inside VisualEyes

10/18/09

This is an overview of the internal design and structure of VisualEyes project scripts. Seeing how the various pieces work together will give you a better sense of the underpinnings that you construct your projects with.

### **What VisualEyes is**

VisualEyes is a Flash-based authoring system that uses the Internet to connect various resources, such as images, maps, video and data together into a seamless interactive presentation. It is a “virtual Lego set” containing a number of features that you invoke by writing a simple script. The VisualEyes Flash web application reads your script and performs the tasks you’ve specified, in much the same way your personal computer reads a program, such as Word and runs it.

The real difference is the “script” that Word was programmed in very complex computer language that takes years to master, while VisualEyes uses a much simpler and easier to learn script based in a format called XML. Because it is programmable, like the computer it is running on, the script VisualEyes runs can be molded to meet the needs of your individual project.

### **What XML is**

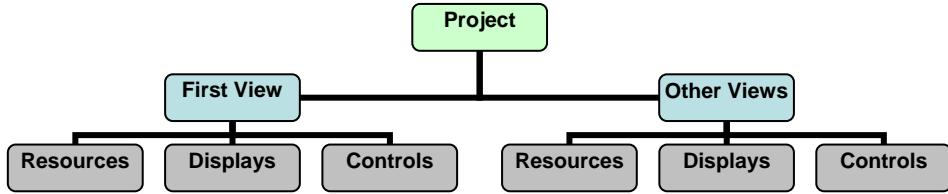
XML (eXtensible Markup Language) has become very popular as a way to request, retrieve, and show data in a webpage. Because it is just plain text, people can write XML in almost any word processor or text editor. It is not a computer language, but more of a storage format for information.

### **VisEdit**

To make project scripts for VisualEyes, you can write the XML directly, but we have developed a tool called VisEdit that hides the formatting issues and lets you concentrate on the functionality of your project. There are also a number of wizards that will guide you through commonly used options.

### **Storage of Resources**

Your project scripts are stored on our server. Creating an account is easy, and provides a secure area to store multiple projects scripts and data files that only you can edit. We do not store images, audio, or video files, so you will need to put them elsewhere, either on a personal website, or a shared storage site such as Flickr or Picasa.



## The Structure of VisualEyes Projects

The topmost level on the hierarchy is the *project*. The project contains a number of *views*. This will allow for multiple interactive representations to be shown simultaneously, making for easy comparisons, with a shared timeline for control.

Each browser may contain any number of views. These views are represented as tabbed areas on the screen. Clicking on any of the tabs will bring up a different view. Each view contains descriptors of the *resource* to display, or use as data to change the display. There can be timeline for each view, which makes it easy to set the temporal aspect of that view.

### Views are made up of resources displays, and controls

Each view contains descriptors of the *resource* to display, or use as data to change the display. There are timelines for each view, which makes it easy to set the temporal aspect of that view, display such as pop-up information boxes and charts as well as control panels to make the view highly interactive.

#### 1. Resources

Resources are the sources of information or media. The resource could be a *map*, some *media*, a table of *data*, or a *graphic*. Each resource item contains a URL where to find the resource and a name or id to refer to it by.

- **Images** - Digitized images of primary source documents from digital archives can be displayed and integrated into maps, animations and other visualizations. These images can be JPEG, GIF or PNG formats, and can be dynamically sized and positioned.
- **Maps and Drawings** - VisualEyes can display scans of historical maps as raster images, vector-based maps from GIS systems such as arcGIS, and vector drawings in Adobe Illustrator's .ai format. All maps can be easily panned and zoomed, with an option to see an overview inset. Shape data can come from specified tables in a database, or a link to an online XML file.
- **Audio / Video Media** - You can seamlessly mp3 audio and Flash video files into projects and have good control of playback, timing, and coordinating actions with the media playback.
- **Data** - A rich array of historical data can be imported into VisualEyes from a database as a table. This data can be displayed as a layer on

a map, shown as a chart, table, or graphic element, and most importantly, used to include or omit events in the views. Data can come from specified tables in a SQL/mysql database, a link to an XML file, or from Internet-based web service such as IBM Research's ManyEyes project.

## 2. Displays

Displays provide the means of showing data and information within your project in the form of pop-up information boxes, charts, paths and other displays.

- **Information boxes** - Information boxes are popup boxes used to display textual information on demand. They are typically called by clicking on path and graph elements. InfoBoxes can contain a kind of HTML formatting and can be populated using search and replace variables that can be set using a database or XML data file.
- **Document readers** - Booklets are a useful way to present information in VisualEyes. Booklets resemble page spreads in traditional books. Booklets contain one or more pages. Each page can have a picture on the left-facing page and text on the right-facing page, a single picture, or text crossing both sides.
- **Charts and Graphs** – Various types of graphs (line, bar pie, scatter, etc.) can be drawn using dynamically generated data base on data from XML or SQL databases, time from timeline, settings of control panel items, or any combination of them.
- **Widgets** - are a type of graph that graphically displays a single continuous value on the screen, such as a dial, clock, thermometer, etc.
- **Paths and Relationship Maps** – A series of positions on the can be defined to appear at particular times. Each position can be marked by an icon, shape or image file. Clicking on the position can call up a web page draw graphical elements, or popup window showing some information. Lines can be drawn to connect these positions. A path can be arrange in a radial concept map format to help visualize relationships between objects shown

## 3. Controls

Controls provide an opportunity for your users to interact with your project with timelines, animation players, control panels, and mini-scripts called GLUE.

- **Timelines** - Each view can have its own timeline that can control the temporal aspect of the project. Sliding the cursor changes the view's

date, which in turn can change the way in which information is displayed if it is time dependent.

- **Animation Players** - The current time on the timeline can be animated over time, using a player control, allowing the project to animate any time dependent elements from any point on the timeline to another.
- **Control Panels** - Views can have multiple pull-out areas docked to a side of the screen that can be expanded or collapsed as needed and contains a number of collapsible check boxes to toggle on and off various features of the map, such as data overlays, roads, town names, etc. Control panels can contain, Radio Buttons, Check Boxes, Combo Boxes, Slider Controls, Text Input Boxes, Buttons, Headers and Legends
- **GLUE** - The General Language to Unite Events is a procedural description of how the various resource elements connect with one another and are displayed. GLUE also contains elements for linking user-generated actions, such as clicking on the screen with actions. GLUE also provides an opportunity to calculate tables and fields in resources based on a simple script in the tag. Many common types of operation can be defined between these elements, so that VisualEyes is able to relate rich data relationships between them and visualize them on a special and temporal basis.
- **Hidden views** – Even though there is room for 8 or less views across the tab area, you can define an unlimited number of hidden views that can be swapped into the tabbed areas, making very complex projects more manageable.

## Using GLUE Scripts

GLUE is used to connect the data resources described above to data consumers such as display tables, popup windows, charts, and data-driven maps. It is written as a series of lines of text in the XML project file. A view may contain any number of lines of Glue and each line has an *id*, or name, so you can string together multiple lines, or cause a line to execute from an item in a control panel.

Glue also provides an opportunity to calculate tables and fields in resources based on a simple script in the tag. Many common types of operation can be defined between these elements, so that VisualEyes is able to relate rich data relationships between them and visualize them on a special and temporal basis.

- **Lists** – These lists are similar to the variables used in programming and Algebra. In VisualEyes, lists may contain one or a series of numbers or words. There are predefined lists that hold things like the current date, position on the timeline, last place clicked, etc.

- **Queries** - Being able to query data without needing to send a request to a server is a big advantage in terms of performance. The queries in VisualEyes are similar to SQL queries in concept and structure. There are 3 parts to querying data: 1). Defining the data set to be queried, 2). Specifying what parts of that data you want to get, and 3). Doing something with the results you get.
- **Math methods** – The usual methods found in math libraries, such as add subtract, min, max, etc. can be used in GLUE scripts.
- **Word replacement** – Much like the old mail merge function in word processors. You can set placeholders in text that can be automatically “found and replaced” with the real information programmatically.
- **Display item filling methods** – There are a number of methods that will fill items such as paths and charts with data from an XML file.
- **Conditionals** - The **if** method will execute the number of lines specified if a particular condition met.
- **Movie control methods** - These methods will control a movie resource's transport functions such as play or stop.

## Summary

By combining these elements and GLUE method, almost any kind of project can be built. We are constantly adding new methods that make commonly used tasks easier to perform by providing new methods that perform larger tasks.

We hope that this has given you a better idea of how to make projects in VisualEyes. Full documentation and a support forum can be found from our website at [www.viseyes.org](http://www.viseyes.org).

Bill Ferster