visualeyes

XML REFERENCE

4/1/11

This reference guide shows all the possible elements that can be used in a VisualEyes project listed alphabetically. The top-most element of any project is the **project** element. All other elements are contained by **project**.

The element titles shown on a grey bar with the element they are contained in written to the right. A short description appears underneath the bar. The element's *attributes* are listed alphabetically with the name, description, list of option (if any) and the default value if the attribute is not specified. If the default value is the one wanted, there is no need to specify it.

A list of elements that may be contained within the element are alphabetically listed and underlined below the attributes list. The element's name is a hypertext link to the reference for that element, and clicking on it will move you to that element.

Project structure:

project	Project start
frame	Defines size/colors of views
logo	Logo for all views
tab	Defines size/colors of view tabs
textformat	Defines size/colors of text
view	Views
cmap	Concept map
dock	Mac OSX-like dock
controlpanel	Control panel for user interactions
glue	Glue scripts
overview	Inset overview to navigate whole when zoomed in
netmap	Organization/network map
path	Path of dots
pmap	Picture-style concept map
resource	Resources of various types
docviewer	An image and text booklet viewer
graph	Graphs and charts
image	JPEG, GIF and PNG Images
infobox	Popup text box
map	Vector map or illustration
movie	Flash movie
xml	XML or CSV file
timeview	Timeview display
table	Table resource
widget	Univariate chart widgets
textformat	Defines size/colors of text
timeline	Timeline control
zoomcontrol	Image zoom control

Resources are listed as separate elements in this document for the sake of clarity, but they are all within a **resource** element with its *type* attribute set to the appropriate type (i.e. *image*, *map*, etc.) These are written with resource in the grey bar (i.e. *image resource*, etc.). A star (*) on the description means that attribute or item is required.

band — timeview

The bands for ***timeviews*** are made up of individual events, each with a date, a label, an icon type, etc, just like dots are used in the ***path*** and ***cmap*** (concept map) displays, and fully clickable.

backImg	Backgound image URL for band	
border	Border amount in pixels	*8*
tickCol	Color tick lines as an RBG hex number	*0x999999*
col	Color of background as an RBG hex number	*0xffffff*
corner	Radius of corner of frame for making rounded rectangles	*0*
frameCol	Color of frame edge as RBG (-1 = none)	-1
hgt	Height of frame in pixels*	
ratio	Percentage of total time to show in band	*100*
dataPos	Position of storylines start: bot | center | top	*center*
tickCol	Color of tick mark lines as an RBG hex number	*0x000000*
tickDateFormat	Date format for tick dates: yr|mo/yr|dy/mo/yr|mo/dy/yr|mo,dy,yr	*yr*
tickDatePos	Position of tick line date text: bot | top	*bot*
tickSpan	Number of days between tick mark lines	*365*
tickWid	Width of tick mark lines in pixels	*0*

dot	Dots(s) in the band*
textformat	Sets default text attributes

cmap — view

Concept maps are similar to paths, but the paths can be arranged in a radial manner similar to a hub and spoke shape. The ***dot***s are not time dependent, and ***line***s (edges) must be specifically drawn by setting the relationships between the ***dot***s (nodes). Labels are automatically drawn if specified underneath the dot.

alpha	Opacity as a number from 0-100	*100*
backCol	Color of interior wash to blot out background as RBG (-1=off)	*-1*
col	Color of line an RBG hex number	*0x00ffff*
cx	Center X position in pixels	
cy	Center Y position in pixels	
hgt	Height in pixels for ovals (omit for perfect circle)	*0*
id	ID of map*	
preload	Load this resource before screen is shown: true | false	*false*
shape	Shape of the concept map: radial	*radial*
stagger	Amount to stagger odd and even spokes of map in pixels	*0*
wid	Width in pixels	*0*

dot	Dots(s) in the map*
frame	Sets box of map*
line	The relationship between the dots and determine how
linestyle	The style of a line
textformat	Sets default text attributes
legend	Adds a legend

controlpanel — view

Control panels provide a dialog box-like means for setting parameters of the screen. These parameters can be set using ***item***s such as check boxes, radio buttons, combo selection boxes, sliders, text input, and buttons to cause some sort of action. Items typically cause some action by adding an id of a **GLUE** element to call when they are changed or clicked.

title	Name of the control panel as it appears in header	
closable	Control panel has closing button: true | false	*true*
open	Control panel is open on startup: true | false	*true*
span	*Column width of for multiple column items*	
title	*Title of panel that appears at top*	*true*

frame	Frame of the project views*
textformat	Overrides view's text format for this panel*
item	Line(s) in the panel

dock — view

A **dock** display presents a series of dots horizontally across the screen in a similar fashion to the application dock used in the Apple Macintosh OSX. The dots are typically icons or images that are fixed to a base bar. As the mouse hovers over one, it and its neighbors grow by the percentage spec'd by the *growth* tag. Setting the *growStyle* to *"single"* will cause only the dot being hovered on to grow while hovered over, as opposed to the default of *"taper",* which also grows the two dots on either side of the one being hovered over as well. The dots can have glue attached to cause some action when clicked. If a *wid* attribute is specified, the number of pictures on the dock will be limited by that number, and green arrows will appear to scroll to additional pictures.

The ***frame*** element sets the bounds of the dock, but since the dock grows and shrinks based on the number of *dots* within it, the dock will draw from the center of area defined by the frame's *left* and *wid* tags. The frame's *hgt* tag defines the height of the base bar. Setting the *hgt* to 0 will inhibit the drawing of the base bar.

alpha	Opacity as a number from 0-100	*100*
growStyle	What pictures grown when moused-over: growth | taper	*growth*
growth	Percentage to grow when moused-over	*200*
id	ID of map*	
preload	Load this resource before screen is shown: true | false	*false*
wid	Limit number of pix by number	*0*

dot	Pictures(s) in the dock*
frame	sets box of dock*

docviewer resource view

An ***docviewer*** is resource very similar an ***infobox*** to that can hold HTML formatted text and a picture side-by-side in series of pages provided by a data source (i.e. and XML file or SQL query).

The data source can have 4 fields: *title, source, desc* and *caption*. The *title* field provides a title at the top and a way to select items from the data source. Items with the same title will appear as pages within the document viewer. The *source* field gives a url for a picture if desired, and *desc* is an html formatted text area. If a *caption* field is defined, it will appear underneath the picture.

If both *desc* and *source* are defined, they will appear side by side. If only one is defined, only that one will appear. The text and picture information is supplied by the filldocviewer() method, typically as the result of a query method. Text can contain the standard HTML formatting macros (see appendix).

arrowpos	Position of page numbers: bot | mid | top	*bot*
border	Border amount in pixels	*24*
close	Has close button: true | false	*true*
id	ID of resource*	
nopan	Inhibit panning on vertical images: true | false	*false*
scroller	Show scroller if text exceeds frame: true | false	*true*
selectable	Text is selectable with mouse: true | false	*true*

page	Page for a document viewer

dot — common

Containers such as paths place ***dot***s in particular places on the screen. A ***dot*** can be a graphic shape, such as a circle or square, an image, or an icon. Dots can have **GLUE** methods associated with them so actions can occur when you click on them. Dots will continue using properties set in previous dots to reduce unnecessary repeating of attributes. For example, if you set the style to triu (up-facing triangle), all dots that follow would be rendered as triu until re-specified.

alpha	Opacity as a number from 0-100	*100*
col	Color of interior as an RBG hex number	*0x00ffff*
date	When dot becomes active (in any date format)	
end	When dot becomes inactive (in any date format)	
frameCol	Color of frame as an RBG hex number	*0x000000*
frameWid	Width frame in pixels	*0*
glue	GLUE id to be called if clicked	
hgt	Height in pixels	*0*
hover	GLUE id to be called if hovered over	
icol	Re-color icon as an RBG hex number	
id	ID of path	
lab	Labels for dot	
labelCol	Color of labels as an RBG hex number	*0x000000*
labelPos	Position of labels relative to dot: bot | center | left | right | top	*bot*
pct	Percentage within the route	
rot	Angle of rotation in degrees	*0*
style	Shape of dot marker (icon: | .jpg | .gif | .swf | .png: bar | but | cir | rbar | span | star | triu | trid | tril | trir)	
time	Time dot becomes active from 0-1	
wid	Width in pixels	*0*
x	X position of dot	
y	Y position of dot	

frame — common

frames are used to define rectangular areas on the screen or size display objects.

alpha	Opacity of frame as a number from 0-100	*100*
backCol	Color of background as an RBG hex number	*0xffffff*
corner	Radius of corner of frame for making rounded rectangles	*0*
docking	Docking mode: left | right | top | bottom | float | center	*float*
dropWid	Width of drop shadow in pixels	*0*
dropBlur	Bluriness of drop shadow (0-9)	*0*
dropCol	Color of drop shadow as an RBG hex number	*0x000000*
frameCol	Color of frame as an RBG hex number	*0x000000*
frameWid	Width frame in pixels	*0*
hgt	Height of frame in pixels*	
left	Number of pixels from left of screen	
top	Number of pixels from top of screen	
wid	Width of frame in pixels*	

glue — view+

GLUE contains scripts that control relationships between resources. See chapter on glue for more information,

from	ID of resource to control	
id	ID name of GLUE script	
init	Run script at each refresh: true | false	*false*
once	Run script at startup only once: true | false	*false*
script	GLUE script code	

gmap resource — view

Embeds a Google map in VisualEyes. The map can be attached to the screen base, much like an ***image*** resource can be, or as a floating window using the *depth* attribute.

depth	If resource is bound to screen: screen | topMost	*screen*
dim	Dimensionality: 2D | orthogonal | 3D	*2D*
frameCol	Color of frame as an RBG hex number	*0x000000*
frameWid	Width frame in pixels	*0*
hgt	Height in pixels	50*0*
id	ID of resource*	
lat	Latitude to center map	*38.14*
left	Number of pixels from left of screen	*0*
lon	Longitude to center map	*-78.45*
maptype	Type of map at start (0=map, 1=hybrid, 2=sat, 3=ter)	*0*
overview	Show overview navigator?: true | false	*true*
top	Number of pixels from top of screen	*0*
type	Type of resource - must be *gmap**	
typemenu	Show map type menu?: true | false	*true*
wid	Width in pixels	*800*
zoom	Show zoom control? true | false	*true*

graph resource — view

VisualEyes supports a number of chart types that can be drawn, including line, area, stacked area, bar, stacked bar, scatter, bubble, picture, and pie charts. If the *radialImg* attribute is set a normal bar chart will be turned into a radial bar chart around the image specified.

backImg	*Backgound image URL*	
border	Border amount in pixels	*24*
close	Has close button: true | false	*true*
depth	Should it always appear on top: screen:topMost	topMost
highWid	Highlight width in pixels	*0*
id	ID of resource*	
legend	Show legend: true | false	*false*
selectable	Text is selectable with mouse: true | false	*true*
radialImg	*URL for background image in radial bar chart*	
radialWid	*Diameter of background image in radial bar chart*	
showValues	Show values on chart (pie): none | percent | true	*none*
subtitle	Sub-title	
stacked	Are data sets stacked atop one another: true|false	*false*
style	Style of chart: area | bar | line | picbar | pie | scatter | area*	*area*
title	Title displayed on chart	

frame	sets box of timeline*
marker	Marker for a chart
textformat	sets default text attributes
xaxis	Defines X-Axis chart settings
yaxis	Defines Y-Axis chart settings

image resource — view

image allow you to add JPEG, GIF and PNG images from any valid URL provided in the *src* tag. These images are added directly to the view's screen (top-left corner by default, but can be anywhere, as set by *top* and *left* tags), where they can be panned and zoomed. Any number of images can be layer. Setting the *depth* to "topMost" will draw the image independent of any panning or zooming. Setting wid to non-zero, sets that image's width to that size.

depth	If resource is bound to screen: screen | topMost	*screen*
frameCol	Color of frame as an RBG hex number	*0x000000*
frameWid	Width frame in pixels	*0*
glue	GLUE id to be called if clicked: glueID	
gb	Bottom georef (i.e 112: 40.0876554)	
gl	Left side georef (i.e 35: -78.00023954)	
gr	Right side georef (i.e 35: -78.00023954)	
gt	Top georef (i.e 112: 40.0876554)	
hgt	Height in pixels (use original if 0)	*0*
id	ID of resource*	
left	Number of pixels from left of screen	*0*
onclick	Glue to call when resource is clicked: glueID	
ondoubleclick	Glue to call when resource is double-clicked: glueID	
onhover	Glue to call when resource is hovered over: glueID	
preload	Load this resource before screen is shown: true | false	*true*
src	Source URL	
title	Title	
top	Number of pixels from top of screen	*0*
type	Type of resource - must be *image**	
wid	Width in pixels (use original if 0)	0

infobox resource — view

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 variant of HTML formatting and can be populated using search and replace variable that can be set using a database. See appendix for text formatting options.e.

backImg	Backgound image URL	
border	Border amount in pixels	*24*
close	Has close button: true | false	*true*
depth	*Should it always appear on top: screen:topMost*	*topMost*
drag	Can drag: true | false	*false*
scroller	Show scroller if text exceeds frame: true | false	*true*
id	ID of resource*	
position	Position of box following click: abs | north | south | east | west	*abs*
selectable	Text is selectable with mouse: true | false	*true*
subtitle	Sub-title	
tail	Box tail if following mouse click: line | none | solid	*none*
title	Title	
[script]	Text for infobox*	

frame — Frame of the box*

intro — project

This will allow you put a dialog box that will enable you to specify a GLUE element to run once at startup, such as an introductory movie,

glue	ID of **GLUE** element in first view to call*	

item — controlpanel

Items in a control panel.

bold	Whether or not text is bold: *true* | *false*	*false*
def	Default value for item on startup: *true* | *false*	*false*
glue	GLUE to be called by item (with optional data)	
id	ID name for control (not usually needed)	
italic	Whether or not text is italicized: *true* | *false*	*false*
labCol	Color of text label	0x000000
linkto	ID of another item that controls this item's visibility	
title	Title that appears by control item*	
type	Type of control (see list below)*	

- **backbuton** A round button with a **<** that will trigger a glue method on click
- **buton** A round button with a **>** that will trigger a glue method on click
- **buttonbar** A square button with the title written inside that will trigger a glue
- **checkbox** A checkbox that will trigger a glue method when clicked.
- **color** A color chip to choose a color from, or type the RGB values
- **combobox** A combo box to choose between several choices
- **half** Used to add a half-space vertically (leading) to the list
- **header** An arrow control to collapse or expand the items that follow
- **legend** Used to put a color choice when drawing legends
- **line** Draws a separator line
- **query** Adds a query line (if: something equals value)
- **radio** A radio button, of which only one is active in a contiguous group
- **search** A text input box with a search button bar attached
- **slider** A horizontal slider to set the value from 0-100
- **text** Displays a line of text

labels — timeline

Defines the format for timeview-like labels under the ***timeline***.

lines	Show lines to labels: true | false	*true*
offset	Distance from time bar to labels in pixels	*8*
pos	Position of labels relative to the main bar: top | bot	*bot*

legend — cmap

Adds a legend

lab	Labels for legends
style	ID of linestyle type

line — cmap

The lines define the relationship between the ***dot***s and determine how they will be placed if in a concept map.

dir	Direction of a line: float | one | two	*one*
from	ID of dot where line is drawn from*	
style	ID of linestyle type*	
to	ID of node where line is drawn to*	

lineStyle — cmap

alpha	Opacity as a number from 0-100	*100*
col	Color of line an RBG hex number	*0x00ffff*
id	ID of path*	
lab	Labels for line	
letter	Letter drawn midway through line in concept maps	
wid	Width of line in pixels	*0*

logo — project

This adds a ***logo*** to display on the screen

left	Number of pixels from left of screen*
source	Filename of logo (including full http:// path and extension)*
top	Number of pixels from top of screen*

map resource — view

Draw a vector map to the ***view.***

depth	If resource is bound to screen: screen | topMost	*screen*
frameCol	Color of frame as an RBG hex number	*0x000000*
frameWid	Width frame in pixels	*0*
glue	GLUE id to be called if clicked: glueID	
gb	Bottom georef (i.e 112: 40.0876554)	
gl	Left side georef (i.e 35: -78.00023954)	
gr	Right side georef (i.e 35: -78.00023954)	
gt	Top georef (i.e 112: 40.0876554)	
hgt	Height in pixels	*0*
id	ID of resource*	
left	Number of pixels from left of screen	*0*
onclick	GLUE id to be called if clicked: glueID	
ondoubleclick	GLUE id to be called if double-clicked: glueID	
onhover	GLUE id to be called if hovered over: glueID	
preload	Load this resource before screen is shown: true | false	*true*
src	Source URL*	
type	Type of resource - must be *map**	

marker — graph

Marker for a chart.

col	Color of marker an RBG hex number	*0x000099*
datawid	width of data (i.e. line or bar)	2
edgeCol	Color of marker edge an RBG hex number, or -1 for none	*-1*
name	Label of marker	
smooth	Are lines or areas curved?	*false*
style	Shape of marker (bar | cir | tri [u | d | l | r])	
wid	Width in pixels	*10*

mysql resource view

Get data from mySQL database.

host	Name of mySQL host*	
id	ID of resource*	
name	Name of mySQL database*	
password	Encrypted password to authenticate*	
preload	Load this resource before screen is shown: true | false	*true*
src	Source URL*	
query	Query to perform*	
type	Type of resource - must be *mysql**	
user	Usename to authenticate*	

movie resource view

Flash video formatted files (.FLV), MP3 audio files YouTube videos and SWF flash files. The *autoplay* tag which determines if the movie playing when it first appears. Omitting the *wid* tag will cause movie and player to size itself to match the native resolution on a Flash movie. Setting the *glue* to some glue object will cause that glue object to be called every *n* ms specified by *time*. *Start* and *end* specify the movies bounds.

autoPlay	Play movie/sound when loaded: true | false	*false*
autoRewind	Rewind movie/sound when finished: true | false	*false*
close	Has close button: true | false	*false*
depth	*Should it always appear on top: screen:topMost*	*screen*
end	Ending time of movie (in ms)*	
glue	GLUE id to be called if clicked: glueID	
id	ID of resource*	
src	Source URL*	
start	Starting time of movie (in ms)	*0*
timer	Time in ms between calls to glue when playing	*250*
type	Must be set to "movie"	*movie*

frame	Frame of the player*

netmap resource view

Network and organization maps are similar to paths, but the dots are arranged according to the *to* and *from* attributes in the ***line*** elements. The ***dot***s are not time dependent, and ***line***s (edges) must be specifically drawn by setting the relationships between the ***dot***s (nodes). Labels are automatically drawn if specified underneath the dot. The ***frame*** specifies the overall bounds of the map.

Setting *shape* to "org" will connect the ***dots*** in squared off lines as in an organization chart. Setting *shape* to "new" will connect the ***dots*** directly as in a network chart. The initial ***dot***'s *from* attribute should be set to "", to connect it to the screen.

Setting the *shape* to "free" will place the ***dots*** according to the dot's *x* and *y* attributes, allowing for free form placement. Lines will connect between the *from* and *to* attributes set in the ***line*** elements

alpha	Opacity as a number from 0-100	*100*
backCol	Color of interior wash to blot out background as RBG (-1=off)	*-1*
id	ID of map*	
shape	Shape of the lines connecting dots: free | org | net	*net*

dot	Dots(s) in the map*
frame	Sets size of map*
line	The relationship between the dots and determines routing*
linestyle	The style of a line
textformat	Sets default text attributes

overview — view

Overview navigation control inset to scroll zoomed screen by.

boxCol	Control box color	*0xffff00*
def	Show on start up: true | false	*true*
docking	Docking location: botLeft|topLeft|botRight|botRight|topRight	*botLeft*
wid	Width of overview	*100*
src	URL of image (full path wih http://)	

page — docviewer

page for a document viewer. The data source can have 4 fields: *title, source, desc* and *caption*. The *title* field provides a title at the top and a way to select items from the data source. Items with the same title will appear as pages within the document viewer. The *source* field gives a url for a picture if desired, and *desc* is an html formatted text area. If a *caption* field is defined, it will appear underneath the picture.

caption	Image caption
desc	Text for description page
src	Source URL for image
title	Page title

path — view

paths place ***dot***s on the screen and can be connected by lines if desired. The width, color, and alpha can be specified. The position of the ***dot***s is set in pixels, relative to the base resource the ***path*** is atop.

alpha	Opacity as a number from 0-100	*100*
col	Color of line, as an RBG hex number	*0x00ffff*
glue	GLUE id to be called if head is clicked	
headCol	Color head as an RBG hex number or -1 to not color icon	*-1*
headEnd	Leave head icon up at end of path: true | false	*false*
headRot	Rotation angle of head icon in degrees	
headSize	Size of head icon in pixels	
headStyle	Image shown at head of path (icon: | .gif | .jpg | .png | .swf)	
id	ID of path*	
res	ID of basemap (only if geo-referencing dot x/y's from lons/lat's)	
showAllDots	Show all dots, regardless of timing: true | false	*false*
tweenLines	Animate line between dots based on timing: true | false	*false*
wid	Width of line in pixels	*0*

dot	Dots(s) in the path*
textformat	Sets default text attributes
pathway	Collections of dots
route	Calls a pathway that contains dot to be draw within a time period

pathway — path

pathways are collections of dots that can be called by routes. The dots specify their time using the pct attribute as a number from 0-1 within the time specified by the ***route***.

id	ID of pathway*

dot	Dots(s) in the pathway*

pointer resource — view

Resource pointers allow you to use a data resource, such as a CSV or XML file from another view without needing to reload the file again in the current view. The resource must be in a previous view to the one you want to add access to, and have an *id* attribute set, so you can identify it using the *src* attribute.

id	*ID of resource to point to in other view**
src	*ID of view where resource was first loaded**
type	*Must be "pointer"**

pmap — view

Picture maps are similar to concept maps, but the ***dot***s can be independently arranged on the screen. The ***frame*** specifies the overall bounds of the map

alpha	Opacity as a number from 0-100	*100*
backImg	Backgound image URL	
id	ID of map*	
preload	Load this resource before screen is shown: true | false	*false*

dot	Pictures(s) in the map*
frame	Frame of the map*
textformat	Overrides view's text format for this display

polygon / polyline — common

Used to define polygons and poly-lines used in maps and other drawings

col	Color of interior as an RBG hex number	*0x00ffff*
edgeCol	Color of edge as an RBG hex number	*0x000000*
edgeWid	Width of edge in pixels	*0*
id	ID of element	
xy	Coordinate data (x,y; … x,y;)	

project

The project is the top-most element of a VisualEyes and holds the various **view**s to display.

title	Name of the project

textformat	Default text format
frame	Frame of the project views*
tab	Defines view tabs*
logo	Logo image for all views
view	Tabbed view(s)

resource — view

Resources contain information to be used by the VisualEyes. This information is most often a table of data, but can be an interactive vector map, text, images, animation, movies, audio, charts, and graphs. The resource tag in the project file provides a way to identify sources and provide named access to the data they contain. This access is useful because once they have been identified; we can refer to them by name later on using lines of **GLUE** to easily create complex visualizations.

The following resource types are available:

graph	Graph or chart
infobox	Information box
movie	Movie
timeview	Timeview display
widget	Widget
image	Image
map	Vector map or drawing
xml	XML/CSV formatted data

route — path

Calls a ***pathway*** that contains ***dot***s to be draw within a time period.

col	Color of line, as an RBG hex number (over-rides path col)
end	Time of the route's end in any time format"
glue	GLUE id to be called when the head icon (if any) is clicked
start	Time of the route's start in any time format*
pathway	ID of pathway containing the dots to draw*

segment — timebar/menubar

Segments are divisions of the ***timeline*** in a ***timebar*** or *menubar* ***widget*** element

end	Time of the segments end in any time format*
glue	GLUE id to be called when segment is clicked
start	Time of the segments start in any time format*
title	Text to be displayed in the segment*

shapedata — common

Used to define shapes used in maps and other drawings.

col	Color of interior as an RBG hex number	*0x00ffff*
edgeCol	Color of edge as an RBG hex number	*0x000000*
edgeWid	Width of edge in pixels	*0*
xOff	Offset from left in pixels	*0*
yOff	Offset from top in pixels	*0*

tab — project

Size and color of a ***view***'s tabs.

curView	The active tab on start up	*1*
hgt	Height of tab	*16*
offCol	Color of tab when inactive	*0xcccccc*
offTextCol	Color of tab text when active	*0x000000*
onCol	Color of tab when active	*0x000000*
onTextCol	Color of tab text when active	*0xffffff*
wid	Width of tab	*100*

table resource — view

This will allow you blank ***table resource*** to the view. That table can have any number of fields. You will typically fill the table by using a query() **GLUE** method.

days	Data column(s) than need date-to-day conversion
id	ID of resource*
src	Names of the fields, separated by | s (i.e. field1 | field2 | field3)*
type	Type of resource - must be *xml**

textformat — common

The various options that a piece of text can have. A ***textformat*** inherits the attributes of any ***textformats*** before so, only the ones that have changed need to be set.

alpha	Opacity of text as a number from 0-100	*100*
align	Alignment of text to the screen: left | right | center	*left*
bold	Whether or not text is bold: true | false	*false*
col	Color of text as an RBG hex number	*0x000000*
font	Font face of text: _sans | _serif | _fixed	*_serif*
italic	Whether or not text is italicized: true | false	*false*
leading	Amount of pixels between lines of text in pixels (0 = leading of 2+size)	*0*
size	Height of text in pixels	*12*
underline	Whether or not text is underlined: true | false	*false*

timebar — timeline

The ***timebar*** element will add a bar to a timeline that will allow the user to set the timeline will show by clicking segments defined by added labels.

all	Add a show all segments button: true | false	*true*
equal	Make all segments equal widths: true | false	*false*
glue	GLUE id to be called when all button is clicked	
hgt	Distance of segments from main timeline	*6*
offCol	Color of inactive segment as an RBG hex number	*0x999999*
offTextCol	Color of inactive segment text as an RBG hex number	*0x444444*
onCol	Color of active segment as an RBG hex number	*0x999999*
onTextCol	Color of active segment text as an RBG hex number	*0xffffff*

segment	Segments are divisions of the timeline in a timebar element

timeline — view

The ***timeline*** will add a graphical timeline that will allow the user to set a time period along a horizontal timeline using a slider bar. A play button can be added to the timeline to animate the setting of the slider bar over time

dateFormat	date format: yr | mo/yr | dy/mo/yr | mo/dy/yr | mo,dy,yr	*yr*
min	Starting time of the timeline in any time format*	
majorTick	Major tick make length in pixels	*0*
minmax	Show values on ends of timeline: true | false	*true*
max	Ending time of the timeline in any time format*	
minorTick	Minor tick make length in pixels	*0*
numTicks	Number of major ticks	*4*
play	Show play button: true | false	*true*
showMinorValues	Show values with major tick marks: true | false	*false*
showValues	Show values with major tick marks: true | false	*false*
sliderDatePos	Show date on slider, or hide it: bot | hidden | none | top	*top*
speed	Speed of playback from 1-100	*50*
start	Initial time of the timeline in any time format on startup	
tickPos	Position of ticks relative to the main bar: top | mid | bot	*bot*

frame	sets box of timeline*
labels	sets the labels under the timeline
textformat	sets default text attributes
timebar	sets punctuated timeline

A ***timeview*** is a display that shows events that are timed to occur at particular dates. It is similar to a traditional graphic timeline like MIT's Simile. A ***timeview*** item can have any number of bands, *each one having it's own time scale*, allowing you to show events that occur in vastly different time scales, such as decades, years and days. All the bands are linked, so scrolling one, scrolls the others.

There are two additional types of timeview, controlled by the *style* attribute. The shelf *style* is used to place dots along a scrollable shelf, and the storyline *style* draws a series of lines that vary up and down.

Setting the *rot* attribute to something other than "0" will cause the timeline's bands to be wrapped around a cylinder in 3D. The cap of the cylinder can be a full oval or cut off at the top with the *capFull* attribute. A double click is required to call the GLUE, if specified.

alpha	Opacity of band background as a number from 0-100	*100*
backImg	Backgound image URL for full frame	
border	Border amount in pixels	*8*
capCol	Color of 3D cap as an RBG hex number	*0x999999*
center	Start dots in center of band: true | false	*false*
close	Has close button: true | false	*false*
dateCol	Color of central date pointer as an RBG hex number	*0x000000*
dateSize	Size of central date pointer	*0*
drag	Can drag timeview box: true | false	*true*
fullCap	Full 3D cap: true | false	*true*
id	ID of resource*	
min	Starting time of the timeview in any time format*	
max	Ending time of the timeview in any time format*	
rot	Angle of 3D rotation (in degrees, 0-45)	*0*
style	Style of display: shelf | storyline | timeview	*timeview*
subtitle	Sub-title	
timeline	Sync to timeline in view: true | false	*false*
title	Title	

band	Band(s) within a TimeView
frame	Sets box of timeview*
textformat	Overrides default text attributes

Each tab in the ***project*** contains a ***view***. The ***view*** contains elements that are displayed on the ***view***'s screen. Resources such as maps, images and data are loaded for display. The scope of any ***view*** is itself, meaning each ***view*** is "an island unto itself."

id	ID of the tab	
pan	Allow panning of screen: true | false	*true*
title	Name of the tab*	
visible	Sets visibility: on | off	*on*

cmap	Concept maps(s)
controlpanel	Control panel(s)
dock	Dock display
docviewer	Add document viewer(s)
overview	Overview navigation control
pmap	Picture Maps(s) for this view
resource	Resource(s)
textformat	Overrides project text format for this view
timeline	Timeline
zoomcontrol	Zoom control for this view

widget resource — view

Widgets are a type of ***graph*** that graphically displays a single continuous value on the screen, such as a dial, clock, thermometer, etc. The range of widgets available will grow with time, but they plot the *val* attribute from *min* to *max.* The magnifier and progress get their size and position from a ***frame*** element. The data is plotted in the color *col*. The title is displayed below the widget except for the dial, where it's in the dial. The value is displayed to 2 decimal places if it is less than 1, or otherwise whole numbers. The size of round widgets like dials look at the *wid* attribute, where things like thermometer use the *hgt* attribute as well.

alpha	Opacity of band background as a number from 0-100	*100*
back	Show dial/clock/spinner background: true | false	*true*
col	Color of marker as an RBG hex number	*0x000000*
glue	GLUE id to be called timer is fired in timer style	
hgt	Height in pixels (not for mag/prog)	*0*
icol	Color of spinner style icon	*-1*
icon	Shape of spinner-style icon: arrow1 | arrow2 | thumb	*arrow2*
id	ID of resource*	
left	Number of pixels from left of screen (not for mag/prog)	*0*
max	Maximum data value	*100*
min	Minimum data value	*0*
src	Source URL	
style	Style: clock | crop | dial | magnifier | menubar | number | | progress | spinner | thermometer	*dial*
title	Title of widget to display	
top	Number of pixels from top of screen (not for mag/prog.)	*0*
val	Initial value to display	*50*
wid	Width in pixels (not for mag/prog)	*0*

xaxis — graph

autoScale	Scale x axis max automatically (scatter charts only): true|false	*true*
col	Color of line as RBG hex number	*0x0000ff*
grid	Show grid lines: true|false	*false*
lab	Labels for data elements	
majorTick	Length of major tick mark in pixels	*0*
max	Maximum data value	
midline	*Draw mid line horizontally : true | false*	*false*
min	Minmum data value	*0*
minorTick	Length of minor tick mark in pixels	*0*
mod	Number to round values by	*1*
title	Title	
showValues	Show numeric values on axis: true|false	*true*
valueCol	Color of values as RBG hex number	*0x0000ff*
valuePrefix	Prefix for value labels	
wid	Length of axis line in pixels	*0*

textformat	Overrides text format for this axis

xml resource — view

This will allow you to added an XML or CSV formatted data file. That file can have any number of fields and rows. The project tool has a converter that takes tab-delineated spreadsheet files and formats it automatically to XML. The actual format is listed in the appendix.

days	Data column(s) needing date-to-day conversion (| separated)	
id	ID of resource*	
preload	Load this resource before screen is shown: true|false	*true*
src	Source URL*	
type	Type of resource - must be *xml**	

yaxis — graph

Defines Y-Axis chart settings.

autoScale	Scale y axis maximum automatically: true|false	*true*
col	Color of line as RBG hex number	*0x0000ff*
grid	Show grid lines: true|false	*false*
majorTick	Length of major tick mark in pixels	*0*
max	Maximum data value	
min	Minimum data value	*0*
minorTick	Length of minor tick mark in pixels	*0*
mod	Number to round values by	*1*
pos	Axis position: left|right	*left*
showValues	Show numeric values on axis: true|false	*true*
title	Title	
valueCol	Color of values as RBG hex number	*0x0000ff*
valuePrefix	Prefix for value labels	
wid	Length of axis line in pixels	*0*

textformat	Overrides text format for this axis

zoomcontrol — view

Zoom control for this ***view.***

def	Starting value of zoom control *(0-10 times)*	*0*
dock	Docking to overview control: *true*|*false*	*false*
left	Horizontal position of zoom control *(in pixels)**	
magnifier	Show magnifier icon: *true*|*false*	*false*
max	Maximum zoom allowed *(1-10 times)*	*3*
top	Vertical position of zoom control *(in pixels)**	