



Maponics™

Theme Customization (User's Guide)

Table of Contents

Introduction – What is TrueVector Free?.....	3
General Behaviour Customization	4
export	4
xmlConfig.....	4
extraConfig	4
General Theme Customization	4
mainBackground	5
symbolizationData.....	5
Dataset Symbolization.....	6
Feature Symbolization.....	7
Summary	8

Introduction – What is TrueVector Free?

TrueVector Free is the Freeware version of Maponics' revolutionary TrueVector Flash Mapping platform. Utilizing static XML files and Adobe Flash technology, a TrueVector Flash Map can provide an interactive web-based mapping application that can zoom all the way to the ZIP Code level, communicating with a database and displaying custom, real-time geographic data.

TrueVector Free provides you with a map of the US that allows your users to select states and capture those selected states in a Javascript function. The ability to respond in the web page to actions taken inside the Flash application enables TrueVector Free to be integrated into existing websites quickly and easily, providing a compelling user experience.

This document describes the various customization options that are available to you, as a TrueVector user, to customize the look and feel of your product. TrueVector comes with a complete suite of recoloring, skinning, theming and branding options that can be as broad as altering every polygonal object displayed, or as detailed as specifying the border color on a single item.

Before reading this document you should:

- Have a standard text editor (not MS Word) ready. A standard editor such as notepad will work perfectly adequately, although we recommend a more advanced editor like jEdit or UltraEdit.
- Be at least familiar with the basics of editing text files and XML.
- Be familiar with HTML-formatted hexadecimal representations of colors – The 6 character RRGGBB notation. ActionScript's RGB values are identical in every way to HTML's RGB values except for one difference – Instead of being preceded by a # sign, they are preceded by the characters 0x. That is, a number 0, not an upper case letter O. In this way, pure blue in HTML, which would be represented by #0000FF is represented in ActionScript by 0x0000FF. **Note: All colors used in the configuration files must be of this format, otherwise TrueVector may not render them correctly.**

All of TrueVector's theme customization options are stored in readily accessible XML files. These are called configuration files. All you need to edit these files is a standard text editor like notepad, although we recommend a more advanced editor like UltraEdit. Changing the way TrueVector looks and feels onscreen is as simple as changing the settings in the XML and reloading TrueVector in your web browser.

TrueVector's theming and colorization can be broken down into three main categories: General Theme Customization, Dataset Symbolization and Feature Symbolization. At first glance, the sheer range of configuration and customization options available to you might seem daunting, but as you discover the flexibility of the system, changing a color will become intuitive.

Please note: Some of the XML nodenames have uppercase letters in them or are numeric. A "standard" XML editor will probably reject the XML as non-compliant. This is true. TrueVector's XML is not designed to be standards compliant, since it is intended to be read only by TrueVector itself.

General Customization Overview

The customization of your TrueVector map is broken down into two stages – Behavior Customization and Thematic Customization. Behavior customization is concerned with the functionality of your TrueVector map – How a user's click interacts with the web page, for example. Thematic interaction is concerned with how the objects within the map actually look.

Behavior Customization is controlled by the tv_cfg.xml file which lives in the same folder as the tvfm.swf file. Thematic Customization is mainly controlled by the theme.xml file, which lives in the /xml folder of your TrueVector installation. For the details of the other thematic customization files, see the General Theme Customization section of this document.

General Behavior Customization

TrueVector's main configuration file is called `tv_cfg.xml` and it should always live in the same folder as the `.swf` file. In TrueVector Free, there are only 3 sections included in this file:

- **export:** This section controls how TrueVector exports the selected list of states to Javascript
- **xmlConfig:** This section simply allows you to turn various objects within the map on or off to suit your particular preference and application.
- **extraConfig:** This section tells TrueVector whether or not to read a thematic configuration file, or to use its default colors and styles.

export

The export section has two options:

- **javascriptCommand:**
Valid values: Any text string.
The string inside javascriptCommand gets passed to the JavaScript function that parses Flash commands for identification.
- **javascriptSeparator:**
Valid values: Any text string.
When TrueVector passes its list of selected states to JavaScript, it will separate this list with the string or character entered here. Standard delimiters are a comma “,” or a colon “:”.

xmlConfig

The xmlConfig section has two options:

- **showCanadaMexico:**
Valid values: “true” or “false”
This option simply tells TrueVector whether or not to show Canada and Mexico as part of the map. If you prefer a “floating” map of the United States, set this to false.
- **showMiscWaterLabels:**
Valid values: “true” or “false”
This option simply tells TrueVector whether or not to show the major water labels surrounding the US. That is, Atlantic Ocean, Pacific Ocean and Gulf of Mexico. If you prefer not to see these, set this to false.

extraConfig

The extraConfig section has just one option:

- **file1:**
Valid values: “xml/theme.xml” or empty.
If you wish TrueVector to use its default colors and styles, leave this option empty.
Alternatively, if you wish to edit the colors and styles contained within theme.xml, leave this option as it is.

General Theme Customization

TrueVector reads its general theme customization instructions from the file `theme.xml`, which in a default setup is placed inside the `xml` folder.

In standard XML fashion, it is broken down into a number of nested sections, which are described below:

- **mainBackground:** This section contains configuration options for the background image.
- **symbolizationData:** This section contains display rules for each type of geometry.

mainBackground

mainBackground has 3 options:

- **scaleBackground**
 - Valid values: true, false
 - Function: Tells TrueVector whether or not to scale the background when zooming in or out. This option has no effect if there is no background image set
- **bgColor**
 - Valid values: Any valid ActionScript formatted hexadecimal RGB value
 - Function: The color of the background. The background is only shown if there is no background image set.
- **bgImg**
 - Valid values: The path, relative to the location of the TVFM .swf file of a jpg, png or gif raster image.
 - Function: The image to load in the background of TrueVector. For best results, it is recommended that this image be the same size as TrueVector itself.

symbolizationData

symbolizationData has 4 main sections, one for each type of object shown – Points, lines, text objects and polygons. Each type of object requires specific display instructions. The options here represent the default display instructions for each type of geometry at each zoom level. Each geometry type section is broken down into zoom level sections, and each zoom level section has its own display rules.

polygonSymbol

polygonSymbols contain display instructions for polygonal objects within TrueVector, such as the major clickable geographic regions like States or Counties. Their options are:

- **fillColor**: The color to display the filled-in part of the polygon. This has a range from 1 to 6 because polygons can potentially have a color ID assigned to them from 1-6.
- **highlightColor**: The color that the polygon's fill changes to when the mouse is moved over it. It is not possible to disable the color changing functionality of TrueVector on a roll over.
- **selectColor**: The color that the polygon's fill changes to when it is selected. Objects that are selected do not respond to the mouse rolling over them.
- **blurColor**: This is the color that the polygon changes to when it is not in focus – That is, when we are at a zoom level other than the one in which this polygon was drawn.
- **strokeColor**: This is the color of the line around the polygonal object.
- **blurAlpha**: This is the opacity of the object when it is not in focus. 1-100, 100 being fully opaque.
- **focusAlpha**: This is the opacity of the object when it is in focus. 1-100, 100 being fully opaque.
- **selectAlpha**: This is the opacity of the object when it is selected. 1-100, 100 being fully opaque.
- **highlightAlpha**: This is the opacity of the object when it is highlighted

pointSymbol

pointSymbols contain display instructions for point objects within TrueVector, such as city dots or interstate shields. Their options are:

- **fillColor**: The color of the symbol, if this symbol is a geometric symbol such as a circle.
- **fillAlpha**: The opacity of the fill of the symbol. 1-100, 100 being fully opaque.
- **strokeWidth**: Currently unsupported. The width of the line surrounding the symbol, in pixels
- **rotation**: Currently unsupported. The rotation, in degrees, of the symbol.
- **symbolType**: The type of symbol to draw. Currently supported types are circle, interstate, highway, star and none.

- **strokeAlpha:** Currently unsupported. The opacity of the line surrounding the geometric symbol. 1-100, 100 being fully opaque.
- **size:** The size of the symbol. This has different meanings depending on the type of symbol.
- **strokeColor:** Currently unsupported. The color of the line surrounding the symbol

optional pointSymbol paramters:

- **points:** Only used for symbolType "star". Number of points of the star
- **innerRadius:** Only used for symbolType "star". Inner radius of the star
- **outerRadius:** Only used for symbolType "star". Outer radius of the star

textSymbols

textSymbols contain text formatting instructions for text objects within TrueVector such as labels. A City point in TrueVector is comprised of both a point symbol and a text symbol, and takes options from both.

- **size:** Size, in pts of the font to use
- **bold:** true or false. Whether or not to bold the text.
- **font:** Any valid Flash font.
- **italic:** true or false. Whether or not to italicize the text.
- **color:** The color of the text.

Dataset Symbolization

In the GIS (Geographic Information Systems) world – the world of digital mapping and cartography, where TrueVector was born – any object that is placed on a map is known as a “feature.” This includes roads, ZIP Codes, bodies of water, street names, anything. In TrueVector, these “features” are grouped into “layers” of a similar geometry type. These layers are what the General Theme Customization controls available to you in `theme.xml` allow you to control.

However, within TrueVector these “layers” are further subdivided by data type. For example, while both a city point and an interstate shield are of the geometric type “point” and as such, both are included in the point layer, they are of different data types and drawn from different datasets. They require different display instructions. To take the most basic example: Normally city points are circles, where as an interstate shield is most definitely not.

So with that in mind, TrueVector also gives you the ability to control the display instructions for geometry types by dataset only. This is a slightly more advanced means of customization and should only be attempted if you are familiar with XML.

The Dataset Symbolization XML, which is what we’re about to edit, is contained in the file `sym_0.xml`. To change the look and feel of an dataset’s features without changing others of the same geometric type, you first need to find which dataset to edit. The names of the datasets used in your project are listed in the `datasets.txt` document which should live in the same folder as this document, the `docs` folder.

Sample `datasets.txt` contents:

```
us_hwy_i_proj_polyline - Data for Interstates shown at State level
us_final_proj_region - Polygonal geometry for US States.
us_county_fake_region - Polygonal geometry for US Counties.
us_topstatecities_point - Cities shown at State level.
```

Therefore, to change the display options for a particular dataset, you need only find the correct dataset for the items you wish to change. A sample `sym_0.xml` is given below:

Sample `sym_1.xml`:

```
<?xml version='1.0' encoding='UTF-8'?>
<symbolization>
  <dataset id="us_misclabels_proj_point">
```

```

    <pointSymbol>
      <symbolType>none</symbolType>
    </pointSymbol>
    <textSymbol>
      <color>0x000000</color>
      <font>Times</font>
      <size>11</size>
      <italic>>true</italic>
    </textSymbol>
  </dataset>
  <dataset id="na_background_clipped_region">
    <polygonSymbol>
      <fillColor>0xFEf1E4</fillColor>
    </polygonSymbol>
  </dataset>
</symbolization>

```

As can be seen, each dataset to be modified is listed in turn, and inside each dataset is placed display information for each of the geometric types of object that that dataset contains. Remember, a label on a map contains both a text symbol object and a point symbol object, and so display instructions for both must be included when controlling the look and feel of labels.

The parameters inside each geometric section are identical to the parameters for the corresponding geometric type outlined earlier in this document. You can either edit the parameters that already exist in the relevant `sym_x.xml` file yourself, or you can create entirely new dataset sections if there are datasets that do not currently have any display information included in the file.

Feature Symbolization

Features, remember, are the individual items drawn on a map. TrueVector also gives you the opportunity to enter individual feature display instructions. This information will be stored in the same file as the dataset-specific display instructions, `sym_0.xml`.

To change the look and feel of an individual feature without changing those of the same geometric type, or even those in the same dataset, you first need to find which feature to reference. Every single item drawn in your TrueVector interactive map has an individual and unique 6-digit ID number. These numbers are held in the geometric XML, so to find the correct dataset, you need to understand a little about how TrueVector's geometric XML is structured.

The geometric XML for TrueVector free is contained in the file `map_000.xml`, which you will find in the `/xml` folder.

A sample of this XML is shown below:

```

<feature oid="100063" state="04" name="Arizona" color="3" ....
<feature oid="100064" state="05" name="Arkansas" color="4" ....
<feature oid="100065" state="06" name="California" color="4" ....
<feature oid="100066" state="08" name="Colorado" color="1" ....
<feature oid="100067" state="09" name="Connecticut" color="2" ....

```

You can see that each state has its own section, and each state has its own id. In TrueVector, we call it oid, for **Object ID**. If you wanted to enter specific display instructions for California, for example, you would use the oid 100065.

A sample entry into `sym_0.xml` to change the look of California would look something like this:

```

<feature id="100065">
  <polygonSymbol>

```

```

        <fillColor>0xFF0000</fillColor>
        <highlightColor>0x00FF00</highlightColor>
        <selectColor>0x0000FF</selectColor>
        <blurColor>0x000000</blurColor>
        <strokeColor>0x000000</strokeColor>
        <blurAlpha>50</blurAlpha>
        <focusAlpha>99</focusAlpha>
        <selectAlpha>99</selectAlpha>
        <highlightAlpha>99</highlightAlpha>
    </polygonSymbol>
</feature>

```

Once again, the parameters entered into the section are the exact same ones for the specific geometry type of the object you want to alter, in this case, a polygon. However, not all parameters are necessary. TrueVector's rendering hierarchy rules work on a per-parameter basis, so you need only enter the parameters that you wish to be different from the default.

You may enter as many feature definitions as you wish into the symbolization xml files, simply list them one after the other like so:

```

<feature id="102929">
    ...
</feature>
<feature id="102929">
    ...
</feature>

```

Summary

In summary, then, TrueVector's customization and theming options are three-tiered, each tier providing a level of control slightly higher than the last, and introducing a corresponding level of complexity.

- General Theme Customization allows you to set the general look and feel of each type of object inside each zoom level.
- Dataset Symbolization allows for a higher degree of control, by giving you the ability to set the look and feel for each object within a specific dataset.
- Feature Symbolization is the most precise level of control available, allowing you to customize the look, feel and behaviour of individual objects.

As you might expect each tier of customization overrides the previous tier, so if the General Theme customization decrees that all polygons are white, then all polygons will be white. However, if the specific Dataset Symbolization instructions read that states are blue, then all polygons will be white, except for states, which will be blue. Then finally, if the Feature Symbolization says that Virginia will be colored black, then all the polygons will be white, except for states, which will be blue, except for Virginia, which will be black.

Editing TrueVector's default theme and colorization scheme can potentially be a complex task. It is complex because it is flexible. However, it need only be as complex as you want it to be. We strive to ship TrueVector with an attractive and functional initial theme, usually echoing your website's own color scheme, or a color scheme previously agreed upon with you, but we do understand that there are times when that may need to change – If you update your website, for example. It is because of those times that we have put a lot of effort into making the colors and theme of your TrueVector Flash Map as configurable and customizable as possible. We hope that you will enjoy experimenting with the thematic configuration options in your TrueVector product and remember – Always make a back up or a copy. If you back up the files before you edit them, you can always return to a last known working state and try again.