TrueVector v2.1
Theme & Design Module
(User's Guide)

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Table of Contents

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby
notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse
engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Introduction

The TrueVector Theme & Design Module allows you to modify almost every cosmetic aspect of your TrueVector map. It is a complete suite of recoloring, skinning, themeing and branding options that can be as broad as altering every single polygonal object displayed, or as detailed as specifying the border color on a single item. The Theme & Design Module also gives you design and theming control over other aspects of TrueVector such as the main user interface menu bar and the design and layout of the individual display windows.

Before reading this document you should:

Have a standard text editor (not MS Word) ready

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 preceeded by a # sign, they are preceeded 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. Since 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. 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.

Please remember, editing these configuration files is entirely optional, and entirely at your own risk. At Maponics, we make every effort to ensure that TrueVector's initial default theme and colorization is attractive and functional before we ship this product to you. Always remember to make a backup of any file you are about to edit. That way, you can replace it easily, should the edits not work the way you expected.

TrueVector's theming and colorization can be broken down into two categories – Map Features and User Interface Objects.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Customizing Map Features

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.
These “features” are then grouped into layers of a similar geometry type. For example, a city point and an
Interstate shield are both point features, and are grouped into a point layer.
Within TrueVector, these “layers” are further subdivided by data type. In the above example, a city point and an
Interstate shield are both of the geometric type point and as such, both are included in the point layer. However,
they are of different data types and pulled from different datasets. They require different display instructions. To
take the most basic example: Normally city points are circles, whereas an Interstate shield is most definitely not.

Types Of Thematic Controls

TrueVector uses the heirarchical nature of GIS datasets to implement a theming structure. Theming and colorization instructions are grouped into “levels”, with instructions in each level overriding the previous level.

There are five levels, with each taking precedence over the previous one.

1. Default instructions
2. Geometry-based instructions
3. Dataset-based instructions
4. Feature-specific instructions
5. Dynamically assigned Two-Way Communication based instructions

Default Instructions

The Default Instructions are hardcoded into your TrueVector Flash Map and cannot be modified in any way. They
provide a very basic base-level set of display instructions for TrueVector. These are the only types of thematic
controls that cannot be modified, however, their presence means that TrueVector always has available to it a very
basic set of rules with which to render any type of geometric shape. They are superseeded by the geometrybased
instruction set.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Geometry Based Instructions

This is the first (and lowest level) instruction set that can be modified. These directives will be found in the file `theme.xml` and contain geometry-based theming commands. This means that there are only three main sets of commands; one for polygons, one for points and one for lines. To assign different directives for different datasets of the same geometry type, you must utilize the dataset-based instruction set

Dataset Based Instructions

The dataset-based instructions allow you to assign theming directives based on individual datasets. This, for example, is where you can give different rules to city points than you do for Interstate shields. These directives live inside a file called `sym_n.xml`, where the n is a number that represents the zoom depth of TrueVector where the features that these rules will apply to would appear. Remember, computers count from 0, so the first level of the map is zoom level 0, and the first file is thus `sym_0.xml`. The instructions that you place in this section will apply to all features within this dataset, so if you wish to give a single feature its own specific display instructions, we move to the feature-specific instruction set.

Feature Specific Instructions

This represents the most detailed level of control that you can have over the look and feel of the objects within TrueVector. These instructions live in the same file as the dataset-based instructions (the `sym_n.xml` files), but take a slightly different format, so that TrueVector knows that they apply to a specific single feature and not to the entire dataset.

Two-Way Communication Theming Instructions

TrueVector's Two-Way Communication Module, which is another optional module available for purchase, is the final and highest level of theming control available. It can send (in real-time) both dataset and feature-specific instructions and the directives sent to TrueVector by this module override any directives found previously in the static configuration files. Since the Two-Way Communication Module enables real-time streaming communication between TrueVector and your backend server, there is no static theming configuration file to edit. If your implementation of TrueVector contains the Two-Way Communication Module, more information can be found in the document TrueVector v2.1 Two-Way Communication Module.pdf.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Geometry-Based Instructions

`theme.xml` is TrueVector's most basic user-editable theming and colorization configuration file. It is broken out
into 3 main sections: mainBackground, symbolizationData and windowParameters. mainBackground and windowParameters are part of the user interface customization controls, and we will return to them later on in this
document.

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

`<lineSymbol>`

lineSymbols contain display instructions for line objects within TrueVector, such as roads. Their options are:

strokeColor: The color of the line

strokeAlpha: The opacity of the line. 1-100, 100 being fully opaque.

strokeWidth: The width, in pixels of the line

`<pointSymbol>`

pointSymbols contain display instructions for point objects within TrueVector, such as city dots or i

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

nterstate
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. Support for
other geometric shapes such as square, star, etc will be supported in future versions of TrueVector
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
`<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 and Feature Specific Configuration

The Dataset and Feature specific symbolization directives are held in the same file. There will be one file for each
zoom level in your map. For example, if you have three zoom levels, these files will be:

- `sym_0.xml`
- `sym_1.xml`
- `sym_2.xml`

Note: If your map is a single-state version of TrueVector, it is highly likely that it initially draws the entire US, and
then pre-zooms to your state. In this case, your initial zoom level is 1, not 0, and the initial file to edit would
therefore be `sym_1.xml` not `sym_0.xml`, because zoom level 0 is *still* the entire US.

Dataset Specific Configuration

tTo 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.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Sample `datasets.txt` contents:

```
geo_2007_01_us_zipsgenthinnedproj_region: US 5 Digit ZIP Code Polygons
geo_2007_01_us_ziplabels_text: US 5 Digit ZIP Code Labels
geo_2007_01_us_3dzipsgen_region: US 3 Digit ZIP Code Polygons
geo_2007_01_us_county_fake_clipped_z_region: US County Polygons
geo_2007_01_us_hwy_u_proj_polyline: US Highway Coordinates at State Level
geo_2007_01_us_topstatecities_point: US City Locations at State Level
geo_2007_01_us_hwy_i_proj_polyline: US Interstate Coordinates at State Level
geo_2007_01_us_final_proj_region: US State Polygons at US Level
```

Therefore, to change the way that all objects within a particular dataset are displayed, you need to find the correct
dataset for those objects. Then, in the correct sym_n.xml file for that zoom level, you would enter the appropriate
configuration directives for the geometry type of the objects inside the dataset you wish to modify.
A sample sym_1.xml file is show on the following page.

Sample sym_1.xml:

```
<?xml version='1.0' encoding='UTF-8'?>
<symbolization>
<dataset id="geo_2007_01_us_shld_st_i_proj_font_point">
<textSymbol>
<size>10</size>
<bold>false</bold>
<font>arial</font>
<italic>true</italic>
<color>0xFFFFFF</color>
</textSymbol>
<pointSymbol>
<fillColor>0x000000</fillColor>
<fillAlpha>100</fillAlpha>
<lineWidth>0</lineWidth>
<rotation>0</rotation>
<symbolType>interstate</symbolType>
<lineAlpha>0</lineAlpha>
<size>1</size>
<lineColor>0x000000</lineColor>
</pointSymbol>
</dataset>
<dataset id="geo_2007_01_us_topstatecities_point">
<textSymbol>
<size>10</size>
<bold>false</bold>
<font>arial</font>
<italic>true</italic>
<color>0x011A20</color>
</textSymbol>
<pointSymbol>
<fillColor>0x011A20</fillColor>
<fillAlpha>100</fillAlpha>
```

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

```
<lineWidth>0</lineWidth>
<rotation>0</rotation>
<symbolType>circle</symbolType>
<lineAlpha>0</lineAlpha>
<size>2</size>
<lineColor>0x000000</lineColor>
</pointSymbol>
</dataset>
<dataset id="geo_2007_01_us_hwy_i_proj_polyline">
<lineSymbol>
<strokeColor>0x105692</strokeColor>
<strokeAlpha>100</strokeAlpha>
<strokeWidth>2</strokeWidth>
</lineSymbol>
</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 parameter names inside each geometric section are identical to the parameters for the corresponding
geometric types 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. As with all theming customizations, you do not need to
supply values for parameters that you wish to remain the same as the default.
For example, if the default configuration for point object labels was this:

```
<textSymbol>
<size>10</size>
<bold>false</bold>
<font>arial</font>
<italic>true</italic>
<color>0xFFFFFF</color>
</textSymbol>
```

And you merely wished to change the font on city labels to be Times, you would need
to write only this in the configuration file:

```
<dataset id="geo_2007_01_us_topstatecities_point">
<textSymbol>
<font>Times</font>
</textSymbol>
</dataset>
```

The objects within the city dataset would be set to the Times font, and all other
parameters would be inheritied from the next level down in the hierarchy.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein. www.TrueVectorTech.com

Feature Specific Configuration

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.

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 unqiue 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 is split up both by zoom level and by parent object. That is, the child regions (for example, counties) of the state "Virginia " have an XML file of their own and the child regions for Amelia county in Virginia (for example, ZIP Codes) also have a file of their own. Inside the file for Amelia County will be the XML necessary to draw every single item that you see when zoomed into Amelia County.

A sample of this XML is shown below:

```
<feature oid="102928" state="51" county="005" name="Alleghany" ...
<feature oid="102929" state="51" county="007" name="Amelia" ...
<feature oid="102930" state="51" county="009" name="Amherst" ...
<feature oid="102931" state="51" county="011" name="Appomattox" ...
<feature oid="102932" state="51" county="013" name="Arlington" ...
<feature oid="102933" state="51" county="015" name="Augusta" ...
<feature oid="102934" state="51" county="017" name="Bath" ...
<feature oid="102935" state="51" county="019" name="Bedford" ...
```

This is an excerpt from the state file (i.e. the file containing the counties) for the state of Virginia. You can see that each county has its own section, and each county has its own id. In TrueVector, we call it oid, for **O**bject **ID**. If you wanted to enter specific display instructions for Amelia County, for example, you would use the oid 102929. Since this is for the counties within the state of Virginia, we are obviously at zoom level 1. Therefore, the file you would edit would be `sym_1.xml`.

A sample entry into `sym_1.xml` to change the look of Amelia County would look something like this:

```
<feature id="102929">
<polygonSymbol>
<fillColor>0xFF0000</fillColor>
<highlightColor>0x00FF00</highlightColor>
<selectColor>0x0000FF</selectColor>
<blurColor>0x000000</blurColor>
<strokeColor>0x000000</strokeColor>
```

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

```
<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. Once again, not all parameters are necessary. TrueVector's rendering 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>
<feature id="102929">
...
</feature>
```

Using the Two-Way Communication Module

If your implementation of TrueVector includes the Two-Way Communication Module, you are able (amongst other things) to send colors for objects to TrueVector in real-time from a server-side script. The Two-Way Communication Module itself is a flexible, complex and powerful addition to your TrueVector product and detailing it's use and operation is beyond the scope of this document. For more information, please read the document titled TrueVector v2.1 Two Way Communication Module.pdf.

When sending theming instructions back to TrueVector as part of the Two-Way Communication module, all instructions are feature-based. There is no provision made for sending geometric or dataset-based configuration directives. The XML that TrueVector expects to be returned will look very similar to the feature-specific configurations within the sym_n.xml files, except that instead of identifying features by oid, the Two-Way Communication module identifies them by a unique string identifier, such as the two-letter state abbreviation, for US States. Below are two examples that demonstrate how the same theming and coloring rules would be applied to New York State, firstly with the Feature Specific static configuration, and secondly by sending the same data via the Two-Way Communication module.

Please note: "123456" may not be the id for New York State in your particular TrueVector implementation. To find

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

the actual ID for New York, you would look inside the map_000.xml file.

Feature Specific Configuration example:

```
<feature id="123456">
<polygonSymbol>
<fillColor>0xFF0000</fillColor>
<highlightColor>0x00FF00</highlightColor>
<selectColor>0x0000FF</selectColor>
</polygonSymbol>
</feature>
```

Two-Way Communication example:

```
<item id="NY">
<polygonSymbol>
<fillColor>0xFF0000</fillColor>
<highlightColor>0x00FF00</highlightColor>
<selectColor>0x0000FF</selectColor>
</polygonSymbol>
</item>
```

For more information, please see the Two-Way Communication Module documentation.

Other Available Thematic Options

There are three other items that TrueVector's thematic configuration allows you to edit:

1. Tooltip Format
2. Embedded Label Format
3. Highlight Type

The configuration directives for editing these items live in theme.xml

Tooltips

Tooltips are the small textboxes that appear when the user moves the mouse over a region. By default they are
disabled on states and enabled on any other zoom levels that don't have display data. To alter this default behavior requires the Action Management Module.

However, it is possible to alter their format and appearance. The directives for controlling tooltip appearance are
grouped within a wrapper `<tooltip>` node that lives inside the `<symbolizationData>` node.

Sample XML:

```
<tooltip>
<active>true</active>
<border>true</border>
<background>true</background>
<backgroundColor>0xFFFFFF</backgroundColor>
<defaultTooltipFormat>
<font>Arial</font>
<bold>false</bold>
<color>0x111111</color>
</defaultTooltipFormat>
</tooltip>
```

The individual configuration directives themselves are fairly self explanatory and, you will note, look somewhat
similar to a the configuration of textfields. This is because, in Flash terms, a tooltip is nothing more than a temporary textfield.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Embedded Label Format

Embedded Labels are the state abbreviations. They are called Embedded Labels because, by and large, they are
"embedded" into the state polygon itself. These directives allow you to control their appearance. They live inside a
wrapper `<defaultEmbeddedLabelFormat>` node inside the `<symbolizationData>` node.
Sample XML:

```
<defaultEmbeddedLabelFormat>
<font>Arial</font>
<bold>false</bold>
<color>0x333333</color>
</defaultEmbeddedLabelFormat>
```

Again, the directives are fairly self-explanatory and simply reflect some simple properties of the textfield that holds
the embedded label text.

Highlight Type

TrueVector's standard behavior is to highlight an object when the user moves the mouse over it. This behavior
can only be changed by purchasing the optional Action Management Module. However, the way the highlight
looks can be altered using the Design Module.
Earlier in this document it was explained how to set the highlight color, using the geometric, dataset-based or
feature-based theming configuration. This directive `<highlightType>`, which lives inside `<symbolizationData>` allows you to alter the way that highlight works.
Node Name: `<highlightType>`
Valid Values: border | fill
If set to border, TrueVector will highlight the current object by highlighting the border of that object using the
current highlight color. If set to fill (default), TrueVector will highlight the current object by changing the fill color of
the object to the current highlight color.

Thematic Summary

TrueVector's customization and theming options for geographic features are multi-tiered, each tier providing a
level of control slghtly higher and more detailed than the last, while 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.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Two-Way Communication (if available) does not offer a higher or more detailed level of control than feature-specific symbolization, but due to it's nature does override all previous levels of command. 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 intial 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.

User Interface Customization

TrueVector's User Interface customization is multi-faceted and flexible. It allows you to:

1. Change the user interface background colors and font color
2. Add a legend
3. Configure any applicable display windows
4. Add a background image
5. Edit the labels that appear on the user interface buttons
6. Edit the labels that appear at each zoom level in the dropdown list.

Some of the User Interface configuration directives live in the standard theme.xml file, however others live in the configuration file ui.xml.

Global User Interface Customizations

These allow you to change globally-affecting elements of the user interface such as the background and text colors. These configuration directives live in ui.xml

```
<color1>
<color2>
<colorPosition1>
<colorPosition2>
<textColor>
<curvedCorners>
<cornerRadius>
```

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

The background of the user interface elements (the menu bar, dialog boxes, etc) is standard across all elements.
It can be (optionally) a gradient of two colors. If you want simply a flat color, set both colors to be the same color.
The gradient is a vertical gradient that runs from top to bottom with position 0 being top and position 255 being
bottom. `<colorPosition1>` is the position (on that 0-255 scale) where color 1 is at 100% intensity. Likewise,
`<colorPosition2>` is the position (on the 0-255) scale where color 2 is 100% intensity. In this way, it can be
seen that many different two-color gradients can be made.
`<textColor>` is, of course, simply the color of the user interface text color. It is also the color of the "previous
geometry" shaped zoom out button.
Some examples:

```
<color1>0xFF0000</color1>
<color2>0x00FF00</color2>
<colorPosition1>0</colorPosition1>
<colorPosition2>255</colorPosition2>
<textColor>0xFFFFFF<textColor>
<color1>0xFFD8D8</color1>
<color2>0xC6C6C6</color2>
<colorPosition1>200</colorPosition1>
<colorPosition2>255</colorPosition2>
<textColor>0xFFFFFF<textColor>
<color1>0x000000</color1>
<color2>0xE2E2E2</color2>
<colorPosition1>0</colorPosition1>
<colorPosition2>25</colorPosition2>
<textColor>0x000000<textColor>
<color1>0x5D5D5D</color1>
<color2>0xE2E2E2</color2>
<colorPosition1>255</colorPosition1>
<colorPosition2>0</colorPosition2>
<textColor>0x000000<textColor>
```

`<curvedCorners>` and `<cornerRadius>` are completely optional and work together. If `<curvedCorners>` is
set to "true", TrueVector looks in `<cornerRadius>` for the value of the radius in pixels. If you do not enter a
value for `<cornerRadius>`, TrueVector will fall back to its default, which is 4 pixels.

Legend Configuration

This section of the configuration allows you to add a static legend to TrueVector. Many TrueVector clients prefer
to place a legend in the encapsulating web page so ave room inside the map window, however, if you wish to
include a legend inside the TrueVector application, these section will allow you to do so.
You can only edit the contents and style of the contents of the legent window with these configuration directives.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

The look and feel of the window itself will follow the global user interface rules as explained in the previous section.
These configuration directives live inside ui.xml.
A sample legend configuration:

```
<legend>
<makeActive>true</makeActive>
<titleText>Legend</titleText>
<bodyText>Number of Subscribers</bodyText>
<keys>3</keys>
<key1>key 1</key1>
<key2>key 2</key2>
<key3>key 3</key3>
<color1>0xFF0000</color1>
<color2>0x00FF00</color2>
<color3>0x0000FF</color3>
<shape1>rect</shape1>
<shape2>circle</shape2>
<shape3>star</shape3>
<body>
<textColor>0x000000</textColor>
<font>Arial</font>
<size>12</size>
<bold>false</bold>
</body>
<title>
<textColor>0x000000</textColor>
<font>Verdana</font>
<size>14</size>
<bold>true</bold>
</title>
</legend>
```

Legend Options:

`<makeActive>`
Valid Values: true | false
This option simply turns the legend renderer inside TrueVector on or off.
`<titleText>`
Valid Values: Any textual string
This option allows you to enter a title for your legend. It is optional.
`<bodyText>`
Valid values: Any textual string.
This option allows you to enter text within the body of the legend. Note: This text is not the text that gets placed
next to each of the individual keys. It is optional
`<keys>`
Valid Values: Any integer.
This option tells TrueVector how many keys there are in your legend.
`<key1>`
Valid Values: Any textual string

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

This is the text that gets placed next to the first symbol in your legend. Note: successive symbols will get their own
nodes, such as <key2>, <key3> etc..
`<color1>`
Valid Values: Any valid actionscript-formatted Hexadecimal color string
This option allows you to set the color of the first symbol in your legend. Note: successive symbols will get their
own color nodes, such as <color2>, <color3>, etc
`<shape1>`
Valid Values: rect | circle | star
This option allows you to set the symbol type of the first symbol in your legend. Note: successive symbols will get
their own shape nodes, such as <shape2>, <shape3>, etc.

Body Parameters

These parameters affect the style of the tex within the body of the legend. This includes both any body text and
individual key text.
`<textColor>`
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
This option determines the color of the text within the body of the legend
`<font>`
Valid Values: Any valid font name
This option determines the font of the text within the body of the legend
`<size>`
Valid values: Any integer
This option determines the font size, in points, of the text within the body of the legend.
`<bold>`
Valid values: true | false
This option detminers whether or not the text within the body of the legend is bold.

Title Parameters

These parameters affect the style of the tex within the title of the legend.
`<textColor>`
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
This option determines the color of the text within the title of the legend
`<font>`
Valid Values: Any valid font name
This option determines the font of the text within the title of the legend
`<size>`
Valid values: Any integer
This option determines the font size, in points, of the text within the title of the legend.
`<bold>`
Valid values: true | false
This option detminers whether or not the text within the title of the legend is bold.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

An Example:

```
<legend>
<makeActive>true</makeActive>
<titleText>Legend</titleText>
<bodyText>Number of Subscribers</bodyText>
<keys>3</keys>
<key1>key 1</key1>
<key2>key 2</key2>
<key3>key 3</key3>
<color1>0xFF0000</color1>
<color2>0x00FF00</color2>
<color3>0x0000FF</color3>
<shape1>rect</shape1>
<shape2>circle</shape2>
<shape3>star</shape3>
<body>
<textColor>0x000000</textColor>
<font>Arial</font>
<size>12</size>
<bold>false</bold>
</body>
<title>
<textColor>0x000000</textColor>
<font>Verdana</font>
<size>14</size>
<bold>true</bold>
</title>
</legend>
```

Display Window configuration

TrueVector comes with a set of standard display windows for performing various tasks that require input from the user and for displaying information to the user: TrueVector's window configuration ability allows you to modify the look and feel of any or all of these windows. Some windows, such as Alert have preset contents that you cannot modify. Some, such as the Display Window are, by their very nature, designed to allow you to edit the content.

1. Dialog Window. This window is used when a question needs to be asked before TrueVector can proceed.

As an example, TrueVector can be configured such that instead of zooming into a county, the user can be asked if all ZIP Codes in that county are to be selected instead. The Dialog Window is the window that handles this interaction.

2. Alert Window. Sometimes TrueVector needs to give a purely informational message to the user without requiring them to make a decision. Examples include informing the user that a ZIP code could not be found when the user has attempted to search for a non-existent ZIP Code, or informing the user that an XML file could not be loaded. The Alert Window is used for this purpose

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

3. Select Window. TrueVector's standard behavior is to allow a user to select a set of items and then export
them to JavaScript. The Select Window is the window that both displays this list of items and gives the user the opportunity to export them.
4. Display Window. Depending on your implementation, you may need TrueVector to display information about an object that a user has selected or is hovering over. The Display Window is used for this purpose. This is the window that you will most likely modify the most.
All the window types have the same modification commands. Like the theming and colorization commands,
TrueVector ships with a full set of default window theming options, so you need only include the directives which
you wish to be different than the default.
These directives will be grouped together inside a different node for each window type, and all the different
window type sections will be grouped together inside a <windowParameters> node.
These directives live inside `theme.xml`
Sample excerpt from theme.xml

```
<windowParameters>
<alert>
<window>
<windowHeight>150</windowHeight>
<windowWidth>100</windowWidth>
</window>
<title>
<titleHeight>30</titleHeight>
<borderWidth>0</borderWidth>
<borderColor>0x000000</borderColor>
<backgroundColor>0xFFFFFF</backgroundColor>
<color>0x000000</color>
<font>Arial</font>
<size>14</size>
<html>false</html>
<bold>true</bold>
</title>
<body>
<frame1>
<background>true</background>
<backgroundColor>0x09427A</backgroundColor>
<scrollbar>none</scrollbar>
<border>false</border>
<color>0xFFFFFF</color>
<font>Arial</font>
<size>12</size>
<html>true</html>
<bold>false</bold>
```

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

```
</frame1>
</body>
</alert>
<display>
<window>
<windowHeight>105</windowHeight>
<windowWidth>150</windowWidth>
<alpha>75</alpha>
</window>
<title>
<titleHeight>30</titleHeight>
<borderWidth>0</borderWidth>
<borderColor>0x000000</borderColor>
<backgroundColor>0xFFFFFF</backgroundColor>
<color>0x000000</color>
<font>Arial</font>
<size>14</size>
<html>false</html>
<bold>true</bold>
</title>
<body>
<frame1>
<background>true</background>
<backgroundColor>0x09427A</backgroundColor>
<border>false</border>
<color>0xFFFFFF</color>
<font>Arial</font>
<size>12</size>
<html>true</html>
<bold>false</bold>
</frame1>
</body>
</display>
</windowParameters>
```

Depending on your TrueVector implementation, you may not have some or all of the available windows. For
example, if your TrueVector map displays data but does not allow region selection, it is likely that you will have
the Display Window, but not the Select Window. Conversely, if your contract states that your users should be able
to select items, but there is no textual data to display, you will have the Select Window, but not the Display
Window.
As can be seen in the example above, this TrueVector map has only Alert and Display windows. Each window
type has its own section within the <windowParameters> wrapper, and each window type's section is broken up

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

into 3 subsections:

1. `<window>` - This is where you define the base window parameters
2. `<title>` - This is where you configure the titlebar of the window
3. `<body>` - This is where you configure the body of the window

Window Parameters

`<windowHeight>`

Valid Values: Any integer

This is the height of the window in pixels. It must include the title height.

`<windowWidth>`

Valid Values: Any integer

This is the width of the window in pixels. TrueVector will attempt to draw the window at this width, however, the minimum width of the window is determined also by the number of buttons along the bottom of the window. For example, the dialog window that asks if a users wishes to select an entire region or zoom into that region has 3 buttons – “Select”, “Zoom” and “Cancel”. The buttons are approximately 100 pixels wide, therefore the minimum width of that dialog window is around 300 pixels. You may make it wider, of course. The best way to decide upon the width of the window is to experiment.

`<alpha>`

Valid values: Any integer from 0 to 100

The opacity of the the entire window.

Title Parameters

`<titleHeight>`

Valid Values: Any integer

The height of the titlebar of the window in pixels.

`<borderWidth>`

Valid values: Any integer

The width of the border of the titlebar in pixels.

`<borderColor>`

Valid Values: Any valid Actionscript-formatted Hexadecimal color string

The color of the titlebar border.

`<backgroundColor>`

Valid Values: Any valid Actionscript-formatted Hexadecimal color string

The color of the titlebar background.

`<color>`

Valid Values: Any valid Actionscript-formatted Hexadecimal color string

The color of the text inside the titlebar

`<font>`

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech' products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Valid Values: Any valid font name
The font of the text inside the titlebar
<size>
Valid Values: Any integer
The size of the font, in points, of the text inside the titlebar
<html>
Valid Values: true | false
Whether or not the titlebar renders simple HTML. This is primarily useful for data sent via the Two-Way Communication module.
<bold>
Valid Values: true | false
Whether or not the text in the titlebar is bold.

Body Parameters

The body of a display window is the area that displays the main text of the display window. As can be seen from
the example above, its directives live inside a wrapper tag named <frame1> Future development of TrueVector
will include multiple frames within each display window, but this ability is not yet supported. The body text lives
within the main window and has a padding of 5 pixels all around it.
<background>
Valid Values: true | false
Whether or not the body text has a background
<backgroundColor>
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
The color of the background of the body text.
<borderWidth>
Valid values: Any integer
The width of the border of the body text in pixels.
<borderColor>
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
The color of the body text border.
<backgroundColor>
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
The color of the body text background.
<color>
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
The color of the text inside the body text
<font>
Valid Values: Any valid font name
The font of the text inside the body text
<size>
Valid Values: Any integer
The size of the font, in points, of the text inside the body text

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

<html>
Valid Values: true | false
Whether or not the body text renders simple HTML. This is primarily useful for data sent via the Two-Way Communication module.
<bold>
Valid Values: true | false
Whether or not the text in the body text is bold.

Configuring The Main Background

There are two options for configuring the background of your TrueVector Flash Map – You can either set the
background color to be a flat color, or you can add a background image.
Adding a background image to your TrueVector map is very simple. The image should be of the same dimensions
as your map – Other dimensions will work, but may not display correctly and should be of the jpg or png formats.
By adding a background image and setting the objects within the map to be slightly transparent some attractive
effects can be generated.
The main background configuration directives are grouped within a <mainBackground> wrapper node and they
live in theme.xml
<bgColor>
Valid Values: Any valid Actionscript-formatted Hexadecimal color string
If you wish your background to be a flat color, simply enter the color value here. Future support for TrueVector will
allow two-color gradient backgrounds.
<bgImg>
Valid Values: The path to a png or jpg image file, relative to the location of the page which contains the Flash
movie.
If you wish to have background image instead of a flat color, you should enter the path to the image here. This
directive overrides the <bgColor> directive.
<scaleBackground>
Valid Values: true | false
Whether or not to scale the background as the map zooms in and out. This is useful if you wish your background
to always be at the same place relative to the map position.

User Interface Button Labels

TrueVector's User Interface configuration allows you to set your own text as the labels on each of the buttons
inside the User Interface. Depending on your TrueVector implementation, you may have any or all of these

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

buttons available to you:

- Search
- Show Selected Items
- Multi-Geography Selector
- Layer Selector

These directives live in `ui.xml`.

Search

Node Name: `<search>`

The search button brings up the search window, which allows your users to type in a ZIP code and search for that
ZIP code.

Show Selected Items

Node Name: `<select>`

This button brings up a window with the list of currently selected regions.

Multi-Geography Selector

Node Name: `<multiGeom>`

This button brings up the Multi-Geography selector, which allows your users to select one of a number of possible
active geographies at that zoom level. Used only with the MultiGeography Module

Layer Selector

This button brings up the window that allows your users to turn on and off various data display layers within the
map, such as highways or cities.

Sample XML:

```
<buttonLabels>
<search>Find It!</search>
<layer>Show/Hide</layer
<multiGeom>Select It!</multiGeom>
<select>Selected Items</select>
</buttonLabels>
```

In the above example, the Search button has be labelled with the text “Find It!”, the Layer Selector button has
been labelled with the text “Show/Hide”, the Multi-Geography Selector has been labelled with the text “Select It!”
and the Show Selected Items button has been labelled “Selected Items”.

Note: The button sizes are fixed, therefore when you choose your text to be displayed, bear in mind the maximum
width of the button.

Dropdown list labels

TrueVector has a standard dropdown list that appears at each Zoom Level and shows a list of the active objects
within that view. By default. this drop down has an initial entry that reads “-- Select One --”. TrueVector’s User

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Interface Customization allows you to change that text to be whatever you wish, depending on the dataset that
contains the currently active objects.
In order to set the drop down text for a particular dataset, you need to first find the correct dataset id. You can
look in the datasets.txt file for this. As mentioned previously, this file contains the ids of all the datasets used
within TrueVector, along with a plain English description of the data contained within each dataset.
The dropdown label configuration directives are grouped within a <dropdownLabels> wrapper node and they live
within the `ui.xml` file.
Sample XML:

```
<dopdownLabels>
<dataset1>
<id>geo_2007_01_us_final_proj_region</id>
<label>Select State</label>
</dataset1>
<dataset2>
<id>geo_2007_01_us_county_fake_clipped_z_region</id>
<label>Select County</label>
</dataset2>
<dataset3>
<id>geo_2007_01_us_zipsgenthinnedproj_region</id>
<label>Select ZIP Code</label>
</dataset3>
</dropdownLabels>
```

As can be seen, simply list each dataset that you wish to specify a dropdown label for in turn and supply the label.
Normally, there will be only one active dataset at each zoom level, but if your implementation of TrueVector
contains the Multi-Geography Module, then you may have more than one dataset active at any given zoom level.
The datasets need not be given in any specific order, so you can simply add additional datasets to the end of the
list.

User Interface Configuration Summary

TrueVector's User Interface customizations are flexible and powerful, easily allowing you to give your TrueVector
Flash Map a unique look and feel as well as ensuring that your users get the best user experience possible. By
allowing you to change the look and feel of display windows and button labels, TrueVector's User Interface
customizations allow you to decide how best to serve your user's needs and communicate the full functionality of
your TrueVector Flash Map

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com

Design Module Summary

With the combination of the Thematic Configuration and the User Interface Configuration, we have tried to give
you the most complete set of look and feel customization available to any Flash Map on the market today. From
simple tasks like changing the overall theme of the polygons to better reflect your corporate look and feel to
detailed tasks such as displaying complex tabular data about a selected area in a way that is clear and easily
readable, the TrueVector Design module gives you complete control over the way your TrueVector Flash Map
looks.

We hope that you will enjoy experimenting with the thematic and user interface configuration options in your
TrueVector product and remember – Always make a back up or a copy of your files. If you back up the files before
you edit them, you can always return to a last known working state and try again.

© 2007 TrueVectorTech. Note: The information contained in this document is privileged and confidential and is protected from disclosure and
forwarding. If the reader of this document is not the intended recipient, or an employee or agent of the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. This document may not be used to reverse engineer TrueVectorTech'
products. TrueVectorTech provides this document as-is and is not liable for any use or misuse of the processes described herein.
www.TrueVectorTech.com