

TrueVector v2.1

Filter Criteria 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

Introduction	3
Overview	3
How does it work?	3
Step By Step Instructions (with example)	4
Step 1: Creating the HTML form.	5
Step 2: Writing the JavaScript function.	5
Step 3: Configuring TrueVector	6
Conclusion	7

Introduction

The TrueVector Filter Criteria Module works in conjunction with the TrueVector Two Way Communication Module to enable your users to further refine the results returned by the Two Way Communication module. This

document explains how the Filter Criteria Module works and how to customize and implement it in your TrueVector implementation.

Before reading this document, you should have:

1. A good understanding of Javascript/HTML and how web pages interact with embedded objects within them, particularly Flash
2. Read the Two Way Communication Module documentation, TrueVector 2.1 Two Way Communication Module.pdf and understood the concepts explained therein about how TrueVector communicates both with the embedding web page and the remote server.
3. A good understanding of XML.

Overview

The TrueVector Two Way Communication module sends a list of the currently visible geographic areas to a

remote script on your server. This then allows you to analyze that list, colorize it and add area-specific details

such as colors and data to be displayed depending on criteria that you have defined previously.

The TrueVector Filter Criteria Module adds to that functionality by allowing your users to enter additional criteria

to filter those returned results by, thus allowing you to further refine them.

An example:

Joe's Realty is a Real Estate company that are using TrueVector to allow their site visitors to search for homes

by geographical area. Joe is currently using the Two-Way Communication Module to color areas based on how

many homes are available in that area. He uses a 3-tier system: Green for over 100 homes, Blue for over 50,

and Red for under 50. Areas that have no homes available are colored Gray (the default).

Joe discovers that his users would prefer to refine the returned numbers by extra criteria, such as homes that

are available with 2 or more bedrooms, or with town water and sewage, or by price. Now he needs to be able to

colorize areas based not only on how many homes in total are in those areas, but also by how many homes in

those areas meet those requirements.

To do this, Joe uses the Filter Criteria Module. He adds a web form to the page that holds the TrueVector Flash

Map and in this form he puts all of the "search by" options that he thinks his customer will want. His users enter

their search criteria and now Joe can tell TrueVector's Two-Way Communication Module to colorize the regions

taking into account the values that the user has selected inside the web form.

© 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

How does it work?

To make TrueVector's Filter Criteria Module work, you first need to put the filter options into a form inside the page that has the map in it. However, when this form's submit button is pressed, it does not submit the form normally, it makes a call to a publically exposed function on the TrueVector flash movie itself, telling TrueVector what options were selected by the user. TrueVector then utilizes its Two-Way Communications module to pass those options to your server-side script (along with the list of currently displayed areas) which can then take the user-selected options into account when deciding what data/colors to return back to TrueVector. This process is real-time and does not require a re-loading of the current region view.

The normal Two-Way Communication process flow looks like this:

fig. 1 – TWC Process Overview Diagram

With the addition of the Filter Criteria Module, the process flow now looks like this:

fig. 2 – TWC + Filter Process Overview Diagram

You can see that in the normal Two-Way Communication process, TrueVector makes a request to the serverside script, which processes that request and returns the appropriate data (colors, textual display data, etc). With the Filter Criteria Module activated, however, the HTML Form can (via JavaScript) pass values to TrueVector before TrueVector makes its request to the server-side script. Those values will then be included in the request that TrueVector makes to the server-side script, and the script can then take them into account when processing TrueVector's request.

Step By Step Instructions (with example)

This section will provide step-by-step instructions for implementing the Filter Criteria Module within your web page and implementation of TrueVector. The code samples given will apply to the fictional example given above of Joe's Realty.

Step 1: Creating the HTML form.

This is just a standard HTML form. However, since you will need to access the values of the elements in this form via Javascript, remember to give each element an id as well as a name. In fact, since you're not submitting

this form normally, you probably don't need the name attribute, but we recommend adding it to ensure W3C HTML compliancy.

The other main difference about this form is that the submit button will not submit the form in the normal way.

Therefore, instead of this:

```
<input type="submit" name="Submit Data" />
```

You will need to write this:

```
<input type="button" name="Submit Data" onClick="doTrueVectorFilter();" />
```

where doTrueVectorFilter() is the name of a Javascript function that will activate the filtering functionality of TrueVector and pass it the data.

Code Sample:

```
<form name="TrueVectorFilter" id="TrueVectorFilter">
<table>
<tr>
<td>Please select desired number of bedrooms</td>
<td>
<select name="beds" id="beds">
<option value="1">1</option>
<option value="1">2</option>
<option value="1">3</option>
</select>
</td>
<td>Please select desired price range</td>
<select name="price" id="price">
<option value="low">Under $100,000</option>
<option value="med">$100,000 ->
$200,000</option>
<option value="high">over $200,000</option>
</select>
</td>
</tr>
<tr>
<td colspan="2"><input type="button" value="Search"
onClick="doTrueVectorFilter();" />
</td>
</tr>
</table>
</form>
```

Step 2: Writing the JavaScript function.

The Javascript function must be able to do 3 things:

1. Read the data from the HTML form and form it into a string that your server-side script can parse and understand.
2. Uniquely identify the TrueVector flash object inside your web page

© 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. Call the `filter()` method on the TrueVector object, passing it the values from the form. When TrueVector receives the call to its `filter()` method, it will immediately make a request to the Two-Way

Communication script on your server, passing the form data exactly as you structured it.

Sample Code:

```
<script>
function doTrueVectorFilter() {
// Initialize this variable. This will hold our string of data to send
var dataToSend;
// initialize this variable. This will be our reference to the flash
object
var tvObject;
// Now read the values of the two form elements
var bedValue = document.getElementById('beds').value;
var priceValue = document.getElementById('price').value
// Now form our string of data. For ease of parsing at the other end,
// we'll structure this like a standard URL query string
// An example of this string would be: beds=2&price=low
dataToSend = "beds="+bedValue+"&price="+priceValue;
// Now we should get a reference to the TrueVector Flash Object.
// This code is taken directly from Macromedia's examples. According
// to Macromedia, you should not use getElementById() here.
if (navigator.appName.indexOf("Microsoft") != -1) {
tvObject = window[movieName]
} else {
tvObject = document[movieName]
}
// Finally, we simply call the filter() method on the TrueVector Flash
object
// and pass it the data string we formed earlier.
tvObject.filter(dataToSend)
}
}
```

Step 3: Configuring TrueVector

When TrueVector receives filter data from your Javascript function, it will make a standard Two-Way Communication request to your server-side script. Since this is a standard Two-Way Communication request, it

will abide by all of the directives inside the Two-Way Communication Module configuration file, `twc.xml`.

Directives for configuring the filter module itself reside inside `filter.xml`, which will live in the `/config` folder with

the other module configuration files, such as `theme.xml` or `twc.xml`. A sample `filter.xml` has been provided for

you with your TrueVector Flash Map distribution.

There are two options available to you that control the behavior of the Filter Criteria Module directly:

```
<filter>
<sendXMLNode>filter</sendXMLNode>
<sendWhenEmpty>>true</sendWhenEmpty>
</filter>
```

The `<sendXMLNode>` node allows you to define the XML node that will contain the filter data; i.e. the data

passed to TrueVector from your JavaScript function that read the HTML form.

© 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 `<sendWhenEmpty>` node only takes two values: "true" or "false". This node simply tells TrueVector whether or not to send an empty node if there has been no filter information passed to it. The request will be identical to a normal Two-Way Communication request in most respects. A normal Two-Way Communication request at US level looks something like this (using the most standard Two Way Communication Module configuration):

```
<xml>
<state>AL:AK:AR ... WA:WY</state>
</xml>
```

A Filter Criteria Module enhanced Two Way Communication request will look like this:

```
<xml>
<filterData>beds=2&price=low</filterData>
<state>AL:AK:AR ... WA:WY</state>
</xml>
```

Note 1. The above example assumes that the user on Joe's Realty site had selected 2 bedrooms and low price and that the Javascript function had formed that into a standard URL query string, as the sample Javascript above did.

Note 2. The name of the node `<filterData>` is entirely configurable by you. This is the only part of the Filter Criteria Module that is configurable.

Sample filter.xml File:

```
<?xml version='1.0' encoding='UTF-8'?>
<filter>
<sendXMLNode>filterData</sendXMLNode>
<sendWhenEmpty>>false</sendWhenEmpty>
</filter>
```

Obviously, the server-side script that TrueVector talks to as part of its Two-Way Communication module will need to be able to interpret the form data added to the TWC request, a task that is outside the scope of this document. However, since it is the server-side script that interprets the data from the HTML form, and not TrueVector, your options for manipulating that data are as endless as your ability to code them. It can also be seen that since it is the server-side script that interprets the form data, writing a "Clear Filter" button is a trivial task.

Conclusion

The Filter Criteria Module, working in conjunction with TrueVector's Two-Way Communication module, allows for far greater flexibility and targeting of the data that you return to your users as they use your TrueVector implementation. It allows you to offer your users far more precise searching to help them find the information they want easier and faster.