Web analytics is about making business sense out of web visitor behavior – the same common-sense analysis that business professionals have been applying to traditional offline business for decades. Gaining the appropriate insight to enable your organization to make smarter business decisions means understanding your site’s business objectives and determining the appropriate web metrics to provide that information.

Today’s business environment typically involves working at remote and off-site locations. Consequently, the people responsible for the web site and those analyzing visitor activity on the web site to measure web metrics often do not work side-by-side. That is, they are frequently in different groups and locations. This situation necessitates coordination between developing and delivering content, and reporting and analyzing the activity results.

To facilitate interaction between departments, locations and individuals, Webtrends provides a feature designed to place the power of analysis into the hands of those individuals who are most interested in it. This feature, Webtrends query parameters, uses typical web site instrumentation to facilitate the analytics end-users want to see.

If you are reading this chapter in the online Help, you can quickly find any parameters of interest by searching the Help. If you want to read the entire chapter, use the Help’s Table of Contents to see all the topics in this chapter.

### How Webtrends Query Parameters Work

By implementing Webtrends query parameters when designing your web site, reporting becomes a part of the web site design process, eliminating the considerable coordination required and reducing the misunderstandings that can happen when the objectives of many groups are at play. Webtrends query parameters are passed in URLs through JavaScript, captured in SmartSource Data Collector log files, and ultimately used by Webtrends Analytics software or Webtrends Analytics On Demand. Webtrends Analytics uses these parameters to analyze your web activity and to produce reports.

### Using Webtrends Query Parameters

If one of your goals is to limit the amount of manual intervention that is required to produce valuable reports, you can implement Webtrends query parameters in your web pages to automate the configuration of many features in Webtrends Analytics, making the Webtrends solutions more insightful with reduced effort.
For example, you can use Webtrends query parameters in HTML META tags to automatically configure page titles, product names and product information, or campaign names. You could also implement a custom “Campaign Submission” page in a company intranet that sends the Webtrends query parameter to automatically configure Webtrends Analytics to report on new campaigns, allowing you to spend less time on system configuration.

Products Using Webtrends Query Parameters

The following Webtrends products take advantage of Webtrends query parameters:

• SmartSource Data Collector
• Webtrends Analytics On Demand
• Webtrends Analytics software
• Webtrends Visitor Data Mart
• Webtrends Visitor Intelligence
• Webtrends Score

Query Parameter Syntax

Webtrends query parameters follow specific syntax that includes name, value, and format. The following sections describe each of these syntax elements

Name Syntax

The name of each Webtrends query parameter uses the following syntax:

Although Webtrends Analytics handles query parameter names without considering case, JavaScript object names are case-sensitive. Therefore, if you want JavaScript tagging to capture events, you must adhere to specified naming conventions. Webtrends reserved query parameters require an upper case name space, either WT or DCS, and the type identifier in lower case.
In addition to the query parameters covered in this chapter, the following name spaces are reserved for Webtrends:

**WT.i**  
Used for product integrations with Webtrends partners.

**WT.z**  
Used by Webtrends Professional Services for customer integrations.

As a best practice, use the following syntax to create custom query parameters:

`DCSext.w_custom_identifier`

For example, you could create a parameter called `DCSext.w_articleid` to track an article ID.

### Value Syntax

The values associated with Webtrends query parameters use the following syntax.

```
Marine;Land%20Vehicle
```

Each Webtrends query parameter name may have one or more values. Some parameters can be specified in pairs or in groups of related parameters. When related parameters have multiple values, these values may be correlated and their position becomes important as shown in the following example:

```
WT.si_n=name1;name2&WT.si_x=position1;position2.
```

In the previous example, if correlation is specified in the report, `name1` is associated with `position1` and `name2` with `position2`.

If there are multiple values, they are typically separated using a semicolon (`;`). You can use other separators, but you must specify the separator in the dimension or measure setting that is based on the parameter.

---

**Note**

Not all Webtrends query parameters support multiple values. For example, HTML Title Page allows only one value.

If the parameter value contains a semicolon, it must be hex-encoded (“%3B”) to differentiate it from the separator.
Numerical Value Format

Unless stated otherwise, numerical values must be specified using the simple US format using the period as the decimal separator with up to 2 decimals and without thousand separators. For example, 12345.67.

Complete Syntax

The Webtrends query parameters are represented as name/value pairs. The name/value pairs adhere to the following syntax:

The Webtrends query parameters are logged in SmartSource Data Collector log files. Each name/value pair is separated by an ampersand (&).

The Webtrends query parameters must co-exist with other well-known parameters. This means that Webtrends query parameters can be mixed with standard well-known parameters. The Webtrends query parameters can be separated from the other well-known parameters using the ampersand delimiter. While a Webtrends namespace is part of each query parameter, it is conceivable that parameter collisions may occur. If a query parameter string contains duplicate key values, the first instance is used and the others are discarded.

Syntax Examples

Single Parameter With a Single Value

The following example shows a page associated with the “Finance Offer” advertising view:

\[WT.ad=Finance\%20Offer\]

Single Parameter With Multiple Values

The following example shows a page associated with both the “Finance Offer” and “FishFinder Offer” advertising views:

\[WT.ad=Finance\%20Offer;FishFinder\%20Offer\]

Related Parameters With a Single Value

The following example shows a page associated with the Campaign “New Product” and the Campaign Event Type “click.”

\[WT.mc_id=New\%20Product\&WT.mc.ev=click\]
Related Parameters With Multiple Values
The following example shows a page associated with the Campaign “Get Results” and “New Product” with the Campaign Event Type “Click.” Note that because the second parameter uses a single value, if correlation is used, the “Click” value of the second parameter is associated with both values of the first parameter.

```
WT.mc_id=Get%20Results;New%20Product&WT.mc_ev=Click
```

## Types of Query Parameters

Webtrends query parameters consist of the following groups:

- **Auto-configuration parameters**
  - Recognized by Webtrends and used to auto-configure certain features. For more information, see “Auto-Configuration Parameters” on page 5.

- **Custom report parameters**
  - Associated with preconfigured custom reports. For more information, see “Custom Report Parameters” on page 11.

- **SmartView parameters**
  - Used by Webtrends SmartView. For more information, see “SmartView Parameters” on page 22.

- **Stored visitor parameters**
  - Used for identifying visitors when the Visitor History database is exported. For more information, see “Stored Visitor Parameter” on page 23.

- **Visitor History parameters**
  - Related to visitor properties added by Webtrends during the analysis process. For more information, see “Visitor History Parameters” on page 24.

- **SDC-generated visitor parameters**
  - Generated by SmartSource Data Collector (SDC). For more information, see “SDC-Generated Visitor Parameters” on page 37.

- **SDC-parameter override parameters**
  - Override SDC parameters on the client side. For more information, see “SDC-Parameter Override Parameters” on page 44.

- **Conversion plug-in parameters**
  - Used by the encoding conversion plug-in. For more information, see “Conversion Plug-In Parameters” on page 46.

- **Content reports parameters**
  - Used to include web 2.0 content information in reports. For more information, see “Content Parameters” on page 20.

## Auto-Configuration Parameters

Webtrends uses certain query parameters to automatically configure the appropriate advanced features and create related reports. The following types of auto-configuration parameters are available:

- Content Group (see “Content Group Parameters” on page 6)
• Marketing Campaign (see “Marketing Campaign Parameter” on page 6)
• Advertising View (see “Advertising View Parameter” on page 6)
• Advertising Click (see “Advertising Click Parameter” on page 7)
• Server (see “Server Parameter” on page 7)
• Scenario Analysis (see “Scenario Analysis Parameters” on page 7)
• Title (see “Title Parameter” on page 11)
• Split (see “Split Parameter” on page 11)

Content Group Parameters
A Content Group definition uses the following parameters:

• WT.cg_n
• WT.cg_s

You can specify multiple content groups per page. The Sub-Content Group parameter is optional. If none of the Content Groups on a particular page contain Sub-Content Groups, Webtrends Analytics may ignore the WT.cg_s parameter. If the WT.cg_s parameter is included, each Content Group value must have an associated Sub-Content Group value. The Sub-Content Group value may be empty.

Visitor Data Mart also uses this parameter. For more information about the complete set of query parameters that Visitor Data Mart uses, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in Webtrends Visitor Data Mart User’s Guide.

WT.cg_n
    WT.cg_n=Name[;…]

This parameter identifies the name of a Content Group. The maximum length is for each Name is 64 bytes.

WT.cg_s
    WT.cg_s=SubName[;…]

This parameter identifies the name of a Sub-Content Group. This parameter is optional. The maximum length is for each SubName is 64 bytes.

Marketing Campaign Parameter
With Webtrends v7.0 and higher, WT.mc_id replaces WT.mc_n and WT.mc_t. For more information, see “Campaign Parameter” on page 18.

Advertising View Parameter
The Advertising View definition uses the WT.ad parameter, which supports multiple Advertising Views per page.

WT.ad
    WT.ad=Name[;…]

The name of the advertisement viewed on a particular web page. The maximum length for each Name is 64 bytes.
Advertising Click Parameter

The Advertising Click definition uses the WT.ac parameter, which supports a single Advertising Click per page. If a page contains multiple ads, you can design the page to respond to clicks so that each click generates a hit to SDC.

WT.ac

WT.ac = Name

The name of the advertisement clicked to reach a particular web page. To capture this information, the Advertising Click must contain an external redirect back to the client. The redirect needs to include the necessary code to generate a hit to SDC or Webtrends On Demand. The maximum length for each Name is 64 bytes.

Server Parameter

The Server definition uses the WT.sv parameter, which supports a single server per page.

WT.sv

WT.sv = Name

This parameter identifies the name of the web server that served the web content. This is used for server cluster load balanced reports. The maximum length for Name is 256 bytes.

---

**Note**

The WT.sv parameter requires that the profile be set up as clustered with at least one server. This is the server that is used if an incoming hit does not explicitly have a server defined.

---

Scenario Analysis Parameters

Scenario Analysis parameters specify well-known paths or processes in your web site and are typically used to measure conversion and abandonment.

In order to report on Scenario Analysis in Webtrends Analytics, you must also specify your Scenario Analysis definitions in Webtrends Administration.

Visitor Data Mart also recognizes these parameters. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User’s Guide.

Scenario Analysis definitions use the following attributes:

WT.si_n – Name of the Scenario Analysis

WT.si_p – Identifies the step by name

WT.si_x – Identifies the step by position

WT.si_cs – Identifies the step in which conversion occurs for Visitor Data Mart profiles.
You can configure multiple Scenario Analyses per page. However, you cannot count the same page as two different steps in the same scenario.

### Note
Because Scenario Analysis dimensions and measures are visit-based, they should only be used with visit-based dimensions and measures. If you create a custom report that uses Scenario Analysis dimensions with a hit-based dimension such as Page Views, the reported page view count will include all the page views of every visitor who saw one of the steps of the scenario, which inflates the actual page view count.

For example, if WT.pn_sku, which is hit-based, is specified on the same page as the Product View scenario step, and you create a custom report using Product SKU as the dimension and four scenario steps as measures, the Product SKU will show a visit for each step of the Scenario, even though it was only specified on one of the steps.

```plaintext
WT.si_n
WT.si_n=Name[;…]
```

Identifies the name of the Scenario Analysis. The maximum length for each Name is 64 bytes. When configuring a new Scenario Analysis definition using Webtrends Administration, make sure that the Scenario Analysis name matches this parameter value.

When specifying multiple Scenario Analysis names, you must also specify multiple step names or step numbers. Each step name or step number is associated with the scenario name having the same position in the list.

```plaintext
WT.si_p
WT.si_p=StepName[;…]
```

Identifies the step by name. When configuring a new Scenario Analysis step using Webtrends Administration, the step name needs to match this parameter value.

```plaintext
WT.si_x
WT.si_x=StepPosition[;…]
```

Identifies the step by numeric position. The value for WT.si_x must be an integer 1 or higher, and values need to be sequential.

### Note
- Webtrends Analytics supports this parameter for Analytics Reports. Use WT.si_x to identify steps for Visitor Data Mart.
- Do not use WT.si_p if you specify the step by position using the WT.si_x parameter.

```plaintext
WT.si_cs
WT.si_cs=1/0[;…]
```

Do not use if specifying the step by name using the WT.si_p parameter. Also, for Visitor Data Mart only, to translate the position numbers displayed in Webtrends Explore into something more meaningful, create a Scenario Analysis Definition in Webtrends Administration. Ensure that you configure a step name for each step.
Identifies whether the page is the step in which conversion occurs when the value is 1. Most likely, this page is the last step in your Scenario.

In addition to this parameter, the web server must also pass the step query parameter for this page. Webtrends Analytics recognizes either \texttt{WT.si\_p} or \texttt{WT.si\_x} for steps. Visitor Data Mart recognizes \texttt{WT.si\_x} for steps.

Multiple values are supported. If multiple values are passed for this parameter and other Scenario event parameters, Visitor Data Mart correlates the values.

Visitor Data Mart recognizes this parameter. However, this parameter is currently not included in any Webtrends Analytics preconfigured custom reports.

### Scenario Analysis Parameters for Shopping Cart Analysis

Use Scenario Analysis parameters to track shopping cart activity. Webtrends Analytics supports these parameters for the purchase conversion funnel report. Visitor Data Mart supports these parameters for multi-dimensional reporting through Webtrends Visitor Intelligence and for visitor segmentation in Webtrends Explore.

Use the following parameter names and values to tag your shopping cart pages. If you use Webtrends Visitor Data Mart or think that you might in the future, use the Step by Number method. If you use Webtrends Analytics, you can use either the Step by Name method or Step by Number method.

<table>
<thead>
<tr>
<th>Step by Name Method (\texttt{WT.si_p})</th>
<th>Step by Number Method (\texttt{WT.si_x})</th>
<th>Query Parameter Purpose and Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{WT.si_n=ShoppingCart}</td>
<td>\texttt{WT.si_n=ShoppingCart}</td>
<td>Identifies the shopping cart scenario. Webtrends Analytics uses the ShoppingCart value to create a purchase conversion funnel report. Visitor Data Mart uses the ShoppingCart value to provide top-level Scenario data.</td>
</tr>
<tr>
<td>\texttt{WT.si_p=CartView}</td>
<td>\texttt{WT.si_x=1}</td>
<td>Identifies the product view step</td>
</tr>
<tr>
<td>\texttt{WT.si_p=CartAdd}</td>
<td>\texttt{WT.si_x=2}</td>
<td>Identifies the cart start step</td>
</tr>
<tr>
<td>\texttt{WT.si_p=CartCheckout}</td>
<td>\texttt{WT.si_x=3}</td>
<td>Identifies the cart check out step</td>
</tr>
<tr>
<td>\texttt{WT.si_p=CartComplete}</td>
<td>\texttt{WT.si_x=4}</td>
<td>Identifies the cart complete step</td>
</tr>
<tr>
<td>\texttt{WT.si_cs=1}</td>
<td>\texttt{WT.si_cs=1}</td>
<td>Identifies the step in which conversion occurs</td>
</tr>
</tbody>
</table>

To collect shopping cart activity for reporting, you need to set up your purchases pages with shopping cart query parameters and create a Scenario Analysis definition in Webtrends Administration. For information about Scenario Analysis definitions, see Administration Help.
To implement shopping cart scenario parameters using WT.si_p:

1. Configure the product detail pages where visitors can click a button to buy a product. These pages differ from the shopping cart basket page that contains all items in the cart.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_p=CartView to identify tagged pages as the first step.

2. Configure the pages where your visitors add or remove items from the shopping cart.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_p=CartAdd to identify tagged pages as the second step in the scenario.

3. Configure the pages where your visitors start the checkout process.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_p=CartCheckout to identify tagged pages as the third step in the scenario.

4. Configure the page where your visitors have successfully completed the purchase:
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_p=CartComplete to identify tagged pages as the fourth step in the scenario.

To implement shopping cart scenario parameters using WT.si_x:

1. Configure the product detail pages where visitors can click a button to buy a product. These pages differ from the shopping cart basket page that contains all items in the cart.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_x=1 to identify tagged pages with the first step in the scenario.

2. Configure the pages where your visitors add or remove items from the shopping cart.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_x=2 to identify tagged pages with the second step in the scenario.

3. Configure the pages where your visitors start the checkout process.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_x=3 to identify tagged pages with the third step in the scenario.

4. Configure the pages where your visitors have successfully completed the purchase.
   a. Use WT.si_n=ShoppingCart to identify tagged pages with your shopping cart scenario.
   b. Use WT.si_x=4 to identify tagged pages with the fourth step in the scenario.
c. If you want to track the purchase complete page as the conversion step for use in Visitor Data Mart, use WT.si cs=1.

---

**Title Parameter**

The Title parameter, WT.ti, supports a single page title per page.

```plaintext
WT.ti = Title
```

The HTML title of the associated web content. If this parameter is found in parameter list, the value is used in the reports. When present, no other page title lookups are performed. The maximum length for Title is 1024 bytes.

WT.ti always overrides other methods of page title lookup. In a case where two different pages (both having WT.ti) end up being rebuilt to the same URL (via URL Rebuilding), the last page seen wins.

**Split Parameter**

The Split parameter, WT.sp, is the default parameter Webtrends uses to split log files for Parent Child profiles. The parameter supports multiple values per page in order for any given page to be part of multiple Child profiles.

```plaintext
WT.sp = ProfileName[;...]
```

The Split parameter is used by the Parent Child profile generator to build child profiles. As a best practice, keep the length of this parameter as short as possible. The format of this value is an alphanumeric string (for example, UTF-8).

**Custom Report Parameters**

This section discusses parameters that are included in a series of preconfigured custom reports. These query parameters are grouped into categories as follows:

- **Search Engine** – see “Search Engine Type Parameter” on page 12.
- **Web Client** – see “Web Client Parameters” on page 12.
- **Transaction** – see “Transaction Parameters” on page 16.
- **Invoice** – see “Invoice Parameters” on page 18.
- **Campaign** – see “Campaign Parameter” on page 18.
- **Campaign Event** – see “Campaign Event Parameter” on page 19.
- **Content** – see “Campaign Event Parameter” on page 19.
Search Engine Type Parameter

WT.srch

WT.srch=SearchEngineType

The Search Engine Type parameter in the query string of the URL from a referring search engine identifies the link as one for a paid search engine phrase.

Webtrends Analytics uses this parameter for Custom Reporting and Visitor History to differentiate a paid search engine from a reference to an organic search engine. Only WT.srch=1 has meaning. No other values are defined or recognized.

---

Note

- Search engine results using WT.srch=1 are limited to only those search engines that are listed on the Webtrends search engine availability list. Using WT.srch=1 will not produce search engine campaign results for search engines that are not recognized by Webtrends.

---

Visitor Data Mart also uses this parameter. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in Visitor Data Mart User’s Guide.

Web Client Parameters

This section describes parameters that are associated with web client (browser) properties. These parameters are typically used for creating custom dimensions that you include in custom reports.

WT.tz

WT.tz=Timezone

Indicates the web client’s time zone and is the offset of the web client from UTC. The value is expressed in hours.

Positive values are given without a sign (WT.tz=2), but negative values require a minus sign (WT.tz=-12). Values can range from -12 to +14.

Example:

```javascript
var dCurrent = new Date();
var tzQueryParam = "&WT.tz=" + escape(dCurrent.getTimezoneOffset());
```

To represent Pacific Standard Time, use the following parameter:

WT.tz=-8

WT.bh

WT.bh=BrowsingHour

Indicates the web client’s local time of day on a 24-hour clock. Values can range from 0 to 23.

WT.ul

WT.ul=UserLanguage

Indicates the web client’s user language.

Example:

```javascript
var ulQueryParam = "&WT.ul=" + navigator.appName == "Netscape" ? escape(navigator.language) : escape(navigator.userLanguage);
```

WT.cd

WT.cd=ColorDepth
Indicates the web client’s screen color depth. It is represented as the number of color bits to which the web client computer’s video display control is set.

Example:
```
var cdQueryParam = "&WT.cd=" + escape(screen.colorDepth);
```

**WT.sr**

Indicates the web client’s screen resolution. It is expressed as the gross width and height of the web client’s monitor. The format of the value is \texttt{width}X\texttt{height} (for example, \texttt{800X600}).

Example:
```
var srQueryParam = "&WT.sr=" + escape(screen.width) + "x" + escape(screen.height);
```

**WT.jo**

Indicates if the web client has enabled Java. Valid values are Yes and No.

Example:
```
var joQueryParam = "&WT.jo=" + navigator.javaEnabled() ? "Yes" : "No";
```

**WT.js**

Indicates whether the web client supports and/or has enabled JavaScript. Valid values are Yes and No.

For example:
```
<SCRIPT LANGUAGE="JavaScript">
... CODE ABOVE WHERE YOU WANT THIS MODIFICATION TO GO... 
var jsQueryParam = "&WT.js=Yes";
... CODE BELOW WHERE YOU WANT THIS MODIFICATION TO GO... 
</SCRIPT>

<NOSCRIPT>
<IMG border="0" NAME="DCSIMG" WIDTH="1" HEIGHT="1" SRC="http://localhost/njs.gif?dcsuri=/nojavascript&WT.js=no">
</NOSCRIPT>
```

**WT.jv**

Indicates the version of JavaScript supported by the web client. If JavaScript is not enabled, this parameter should not be included.

Example:
```
var jvQueryParam = "";
<SCRIPT Language="JavaScript 1.0">
jvQueryParam = "&WT.jv=1.0"
</SCRIPT>

<SCRIPT Language="JavaScript 1.1">
jvQueryParam = "&WT.jv=1.1"
</SCRIPT>
```

---

**Note**

You can use Webtrends to set up a custom reports (augmenting the Pages reports) that keys on \texttt{WT.js} and provides a targeted count of no JavaScript hits.
WT.ct = connectiontype
Identifies the visitor’s connection type. You can use this parameter in custom reports to determine whether visitors can download media on your site that requires high-bandwidth connections.

The JavaScript tag generates this value using data from Microsoft Internet Explorer 5 or higher. Valid values are lan, modem, and offline. For all other browsers, the JavaScript tag generates a value of unknown.

WT.hp = isHomePage
Indicates whether your visitors have configured a URL as their home page. It is only available for visitors using Microsoft Internet Explorer 5 or higher. Valid values are 1 and 0.

You can use this parameter to filter a report based on URLs or pages, and report on pages that are used as home pages by visitors.

WT.bs = browserSize
Identifies the actual size of the visitor’s browser window. It is expressed as the width and height of the web client window in pixels. The format of the value is widthxheight (for example, 924x212).

WT.fi = isFlashInstalled
Indicates whether your visitors have installed the Macromedia Flash browser plug-in. Valid values are Yes and No.

For more information see, the WT.fv parameter.

WT.fv = flashVersion
Indicates the version of the Macromedia Flash browser plug-in if installed. Requires the presence of WT.fi = Yes.

web. For example, WT.fv = 7.0.

WT.le = languageEncoding
Specifies the character set used by the web client to render the current document. This parameter can be used to troubleshoot internationalization issues. The format of this value is an alphanumeric string (for example, UTF-8). Go to the following site for a list of valid values:

http://www.iana.org/assignments/character-sets

WT.mle = metaLanguageEncoding
Specifies the character encoding set by the web client to render the current document. If the page includes the http-equiv="Content-Type" META tag, the Webtrends JavaScript tag generates this parameter and passes it as an alphanumeric string (for example, UTF-8). The parameter is used for troubleshooting internationalization issues.

For example, the JavaScript tag generates a value of WT.mle = UTF-8 for the following tag:
Go to the following site for a list of valid values:
http://www.iana.org/assignments/character-sets

**WT.em**

WT.em=esc/uri

Specifies the encoding method supported by the web client. This parameter can be used to troubleshoot internationalization issues. Valid values are **esc** for the JavaScript `escape()` function and **uri** for the JavaScript `encodeURIComponent()` function.

**WT.slv**

WT.slv=Silverlight_version

 Specifies the version of the Microsoft Silverlight plug-in installed on the visitor’s web client. If Silverlight is not enabled, this parameter is not provided.

**Products Parameters**

You can use Products parameters to report on transaction activity for products on your site. Several preconfigured custom reports use these query parameters.

These parameters support multiple values except where noted. When multiple values are passed, the order of the values is important because they correlate to other parameters. For example, multiple product values for the **WT.pn_sku** parameter correlate to the number of units passed in the **WT.tx_u** parameter.

**Note**

You cannot use these query parameters as dimensions and measures in Scenario Analysis funnels. Products parameters are hit-based while scenario parameters are visit-based and should not be used together in reports.

**WT.pn_sku**

WT.pn_sku=productSKU[;...]

Identifies the SKU (a unique numeric identifier) of the product. Visitor Data Mart also uses this parameter. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in Visitor Data Mart User’s Guide.

**Note**

**WT.pn_sku** replaces the **WT.pn** and **WT.pc** parameters. **WT.pn** represented the name of the product. **WT.pc** represented the category of the product.

**WT.pn_id**

WT.pn_id=productId[;...]

Note
Optional: Identifies the product identifier of a product. If possible, product IDs should be unique values to preserve lookup data integrity. The following parameters are automatically added by the product translation process when the product SKU is found in the product translation file. For more information about translation files, see “Using Lookup Tables for Analytics Reports” in the Administration User’s Guide.

**Note**

Product IDs typically map to multiple Product SKUs. For example, a sporting goods company might have an item with a specific ID and several SKUs corresponding to various colors.

---

**WT.pn.fa**

\[WT.pn.fa=productFamily[; ...]\]

Identifies the family of the product.

**WT.pn.gr**

\[WT.pn.gr=productGroup[; ...]\]

Identifies the group of the product.

**WT.pn.sc**

\[WT.pn.sc=productSubCategory[; ...]\]

Identifies the sub-category of the product.

**WT.pn.ma**

\[WT.pn.ma=productManufacturer[; ...]\]

Identifies the manufacturer of the product.

**WT.pn_su**

\[WT.pn_su=productSupplier[; ...]\]

Identifies the supplier of the product.

---

**Transaction Parameters**

**WT.tx_t**

\[WT.tx_t=Type[; ...]\]

In earlier versions of Webtrends, this parameter was used to identify transaction types. Because Webtrends Analytics does not use this query parameter in any preconfigured custom reports, it is no longer supported.

**WT.tx_u**

\[WT.tx_u=Units[; ...]\]

Identifies the quantity in the transaction. Pass a positive integer for this value.

If an order contains multiple products, separate the numbers of units for each product using a comma or semi-colon (configurable) in the WT.tx_u variable.

When associating this measure with the product dimension in a Webtrends Analytics custom report, make sure you select the correlation option to ensure that the first number of units is associated with the first product, the second number of units with the second product, and so on.

Visitor Data Mart also recognizes this parameter. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User’s Guide.
**WT.tx_s**

**WT.tx_s=Subtotal[;…]**

Identifies the total cost for each product in the order.

The format of this field must match the currency format that Webtrends Analytics is configured to analyze. Do not include a currency symbol and be sure pass the value in *dollars,cents* format. For example, if you globally define your currency as US-Dollars in Webtrends Analytics, the format of this parameter set to a $4500 cost is: **WT.tx_s=4500.00**. Do not include a currency symbol or a comma in the value.

If an order contains multiple products, the totals for each product should be separated by a comma or semi-colon (configurable) in the **WT.tx_s** variable.

When associating this measure with the product dimension in a custom report, make sure you select the correlation option to ensure that the 1st amount is associated with the 1st product, the 2nd amount with the 2nd product, and so on.

Visitor Data Mart also recognizes this parameter. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User’s Guide.

**WT.tx_e**

**WT.tx_e=event**

Identifies a product-related event. Webtrends Analytics uses this parameter as qualifier in preconfigured measure definitions in order to determine which product to count for a measure. Webtrends Visitor Data Mart uses this value to qualify certain preconfigured events. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User’s Guide. You can use the parameter to report on specific activities by creating a custom report filter that you apply to a custom report.

---

**Note**

Even if there are multiple values specified in the **WT.pn_sku (Products)**, **WT.tx_u (Units)** and **WT.tx_s (Revenue)** parameters, the **WT.tx_e** should contain a single value (the same event applies to the entire page view).

---

You can pass a custom value for this parameter, or if you want to track product purchases, product views, product cart additions, and product cart removals, pass one of the following values:

**WT.tx_e=p**

Required by Webtrends Analytics and Visitor Data Mart to identify a product purchase. Although Webtrends Analytics does not use **WT.tx_e=p** to determine whether a visitor is a buyer (**WT.vr_brws**), Visitor Data Mart does. Instead Webtrends Analytics uses **WT.tx_s** to determine whether the visitor is a buyer. In addition, you can use the Invoice query parameters to include invoice number, date, and time for the purchase. For more information, see “Invoice Parameters” on page 18.

**WT.tx_e=v**

Required by Webtrends Analytics and Visitor Data Mart to identify a product view.

**WT.tx_e=a**

Identifies a product cart addition.

**WT.tx_e=r**

Identifies a product cart removal.

**WT.tx_cartid**

**WT.tx_cartid=CartIdentifier**
Pass a unique value to identify a visitor’s cart. Visitor Data Mart uses this parameter to identify events associated with a specific cart. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the *Visitor Data Mart User’s Guide*.

### Invoice Parameters

Use WT.tx_i, WT.tx_id, and WT.tx_it parameters together. You must enable Visitor History in order to use the Invoice parameters.

**WT.tx_i**

WT.tx_i=InvoiceNumber

Identifies the invoice number for the purchase. Webtrends Analytics uses data stored in Visitor History to make sure that a page view with an invoice number is a new purchase and not the result of a visitor refreshing the page after making a purchase. If Webtrends Analytics sees a page view with an invoice number, that page view is compared against the last three invoices for a visitor. If the WT.tx_i value does not match the last three invoices, Webtrends considers it a new purchase.

Visitor Data Mart also recognizes this parameter. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the *Visitor Data Mart User’s Guide*.

---

**Note**

Webtrends Express Analysis cannot use the invoice data to identify duplicate purchase records because the Express Analysis Engine does not use visitor history data.

**WT.tx_id**

WT.tx_id=InvoiceDate

The format is mm/dd/yy. A 4-digit year is also allowed: mm/dd/yyyy.

Identifies the purchase invoice date, which is based on GMT. If the invoice date is three days older than the date of the visit, it is assumed that the hit is not an actual purchase but a view of a previously bookmarked or saved page. If the invoice date was less than three days than the date of the visit, the WT.tx_i parameter is used to determine if the hit is a valid purchase.

For example, a visitor makes a purchase. The purchase is accounted with an invoice date. The visitor saves a bookmark to the page. Five days later, the visitor goes to the bookmarked page. This causes another hit to be sent to SDC. However, the WT.tx_id parameter still contains the original purchase date. Webtrends analysis sees that the date of the hit is several days after the date found in the WT.tx_id parameter and determines that this is not an actual purchase.

**WT.tx_it**

WT.tx_it=InvoiceTime

Identifies time of the invoice. The format is hh:mm:ss where hh is in a 24-hour format (0 = midnight, 23 = 11pm).

This parameter helps determine when an invoiced purchase was made. This value is used along with WT.tx_id and WT.tx_i to determine if the purchase was a valid purchase or if this was a user refreshing the web page after a purchase or returning to the page to check status.

### Campaign Parameter

**WT.mc_id**

WT.mc_id=Campaign ID

---
Identifies a specific marketing campaign. Pass this query parameter to pages that you want to associate with a specific campaign. You can specify a numeric or string value.

**Webtrends Analytics Considerations**

If you plan to export Webtrends Analytics report data to a SmartReport, values that are either characters or a combination of numbers and characters work best with Microsoft Excel.

If you enable Visitor History in a profile, Webtrends Analytics reads this parameter and stores it in the Visitor History database. The most recent value of this parameter is made available by Visitor History on every hit as `WT.vr.rac`.

**Webtrends Visitor Data Mart Considerations**

Visitor Data Mart uses this parameter to identify Ad events. For more information about the parameter set used by Visitor Data Mart, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User's Guide.

---

**Note**

`NWT.mc_id` replaces the `WT.mc_n` and `WT.mc_t` auto-configuration parameters in earlier versions of Webtrends.

---

**Campaign Event Parameter**

`WT.mc_ev`  
`WT.mc_ev=EventType`

This parameter identifies an ad event type.

Webtrends Analytics and Visitor Data Mart recognize the following values:

- `WT.mc_ev=click` identifies an ad clickthrough event.

**Webtrends Analytics Considerations**

Webtrends Analytics does not currently provide preconfigured custom reports that use the `click` value. However, you can also specify a custom value to identify a custom event type, create a custom report filter based on this value, and apply it to a custom report.

**Segment Parameter**

`WT.seg_X`  
`WT.seg_X=Segment`

Identifies a segment of interest. `X` can be 1, 2, 3, or 4. For example, `WT.seg_1=Segment1`. This parameter identifies values associated with this segment, and you can store these values in Visitor History. The Visitor History function recognizes these values and stores them in the Visitor History database. For more information, see “Visitor Segmentation Parameters” on page 37.

**Page of Interest Parameter**

`WT.pi`  
`WT.pi=Page identification`

---
This parameter identifies a page on your site that is critical to evaluating performance. *Page identification* can be any string. When you enable **Page of Interest Unique Visitor Tracking** in the Visitor History dialog of a profile, Webtrends stores the values for the *WT.pi* parameter for each unique visitor in the Visitor History database. You can limit the amount of disk space used to store these values by keeping the strings as short as possible.

As a best practice, you should only identify key pages on your site with this parameter because Webtrends stores a maximum of 20 pages for each unique visitor for each profile.

Use this parameter to create a Page of Interest dimension that can be associated with a measure based on the *WT.vr.piv* Visitor History parameter. For more information, see *WT.vr.piv* on “Visitor Tracking Parameters” on page 35.

**On-Site Search Parameters**

On-site search parameters allow you to collect activity about your on-site search tool.

**WT.oss**

*WT.oss=Search phrase*

Identifies a word or a phrase that visitors submit for an on-site search.

**WT.oss_r**

*WT.oss_r=number of results*

Identifies whether or not an on-site search is successful. This parameter should be specified on the same hit as WT.oss and should be set to the number of results whenever the on-site search is successful, or to 0 when the search fails (no result).

Webtrends Analytics uses this parameter in preconfigured custom report filters. Webtrends Visitor Data Mart uses it provide data for the Number of Results attribute.

**Registered Visitor Parameter**

**WT.rv**

*WT.rv=1*

If you use Visitor Data Mart, your web server should pass this parameter with a value of 1 when a visitor has completed a registration process. For more information, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the *Visitor Data Mart Administration sGuide*.

**Content Parameters**

The parameters defined in this section can be used to populate custom reports that include web 2.0 content information.

**WT.rss_f**

*WT.rss_f=FeedName*

This parameter identifies the RSS subscription feed. To indicate that a feed read request was made for an RSS feed, use this parameter with *WT.rss_ev=f*.

*WT.rss_f=Sports & WT.rss_ev=f* indicates that the RSS feed name *WT.rss_f* = “sports” was the object of an RSS feed request event.

To indicate that a subscription request was made for an RSS feed, use this parameter with *WT.rss_ev=s*. 
For example, `WT.rss_f=News&WT.rss_ev=s` indicates that the RSS feed name “News” was the object of an RSS subscription event.

`WT.rss_a`  
`WT.rss_a=ArticleName`

This parameter identifies the RSS article. Use this parameter with `WT.rss_ev=a`.

`WT.rss_a=Global%20shortage%20of%20flu%20vaccine&WT.rss_ev=a` indicates that the RSS article name “Global shortage of flu vaccine” was the object of an article read request event.

`WT.rss_ev`  
`WT.rss_ev=a` or `f` or `s`

This parameter identifies the RSS related event that has occurred, such as an article request `WT.rss_ev=a`, feed read request `WT.rss_ev=f`, or subscription `WT.rss_ev=s`.

`WT.clip_t`  
`WT.clip_t=MediaType`

This parameter identifies the type of media that users have accessed. Use this parameter with `WT.clip_n=` and with `WT.clip_ev=`.

`WT.clip_t=Windows%20Media&WT.clip_n=Milton%20Waddams%20Presents&WT.clip_ev=v`

In this example, the clip type `WT.clip_t=` is “Windows Media,” the name of the clip `WT.clip_n=` is “Milton Waddams Presents,” and the event type `WT.clip_ev=v` is view.

`WT.clip_n`  
`WT.clip_n=MediaClipName`

This parameter identifies the name of the clip `WT.clip_n=` that users have accessed. Typically, use this parameter with event type, `WT.clip_ev=`, and media type `WT.clip_t=`.

`WT.clip_ev`  
`WT.clip_ev=EventType`

This parameter identifies the type of media-related event that has occurred, such as a view of a media event, `WT.clip_ev=v`.

`WT.ria_a`  
`WT.ria_a=ApplicationName`

This parameter identifies the name of the Rich Internet Application (RIA) accessed. Typically, use this parameter with `WT.ria_c=` to identify RIA content, `WT.ria_f=` to identify the RIA feature, and `WT.ria_ev=` to identify the RIA event type.

`WT.ria_a=Homepage%20interactive%20promo%20with%20video%20and%20mp3&WT.ria_c=Vegas%20Dvd%20Video%201&WT.ria_f=Play&WT.ria_ev=play`

In this example, the parameter identifies the RIA application as “Homepage interactive promo with video and mp3,” the RIA content as “vegas,” the RIA feature used as “play,” and the RIA event type as “play.”

`WT.ria_c`  
`WT.ria_c=RIAContent`

This parameter identifies the RIA content, `WT.ria_c=`. Typically, use this parameter with `WT.ria_a=`, `WT.ria_f=`, and `WT.ria_ev=`.

`WT.ria_f=`  
`WT.ria_f=RIAFeature`
This parameter identifies the RIA feature accessed, \( WT.\text{ria}_f = \). Typically, use this parameter with \( WT.\text{ria}_a = \), \( WT.\text{ria}_c = \), and \( WT.\text{ria}_ev = \).

\( WT.\text{ria}_ev = \)
\( WT.\text{ria}_ev = \text{RIAEvent} \)

This parameter identifies the RIA event that has occurred, such as the selection of a button or feature that is part of the RIA application used. Actions taken in the RIA application such as “play,” “zoom,” or “spin” are examples of an RIA event.

\( WT.\text{cgm}_t = \)
\( WT.\text{cgm}_t = \text{CGMType} \)

This parameter identifies the type of consumer generated media, \( WT.\text{cgm}_t = \). Use this parameter with \( WT.\text{cgm}_ev = \).

\( WT.\text{cgm}_t = \text{Blog} & WT.\text{cgm}_ev = c \)

In this example, the parameter identifies the consumer generated media type as “blog,” and the event \( WT.\text{cgm}_ev = c \) is a comment.

\( WT.\text{cgm}_ev = \)
\( WT.\text{cgm}_t = \text{CGMEvent} \)

This parameter identifies the consumer generated media event that has occurred, such as a post (\( p \)) or comment (\( c \)).

\( WT.\text{test}_v = \)
\( WT.\text{test}_v = \text{Variant} \)

This parameter identifies the test variant, \( WT.\text{test}_v = \) for the web site. This parameter can be populated then used to compare activity such as the scenario step conversion rate of two or more test variants.

\( WT.\text{test}_v = \text{Product Layout B} \)

In this example, the web site test variant used is “Product Layout B.”

### SmartView Parameters

This section contains parameters defined specifically for use with Webtrends SmartView. For information about configuring Webtrends for SmartView, see the *SmartView User’s Guide*.

\( WT.\text{svl} = \)
\( WT.\text{svl} = \text{any string} \)

For SmartView to differentiate multiple links on a web page that all lead to the same URL, use the \( WT.\text{svl} = \) query parameter to uniquely identify the links. For example, if you have two links on your home page that both go to the store page, you should use the SmartView query parameter to identify each link.
To use the SmartView query parameter:

1. Place the WT.svl parameter on every page where multiple links lead to the same page. For example, http://www.mydomain.com/?WT.svl=link1.

2. Assign each link a unique value. SmartView uses WT.svl to assign the appropriate measure values to individual links.

---

Notes

Do not include this parameter in your URL Rebuilding definitions. Webtrends automatically recognizes this parameter and uses it only when creating SmartView custom reports. If you include WT.svl in a URL Rebuilding definition, non-SmartView reports are affected. If you exclude it in a URL Rebuilding definition, Webtrends won’t be able to use it to differentiate links when creating SmartView reports.

---

WT.tsp

WT.tsp=1

Identifies transition source pages for SmartView. The JavaScript creates and passes this query parameter if you enabled SmartView page transition tracking in the JavaScript tag and you tagged the source page with the SmartView page transition META tag. A source page is a page that you want to be tracked for SmartView reporting. For more information, see “Configuring SmartView Using JavaScript Tags” in the SmartView User’s Guide.

Do not use this query parameter to tag your web site. You can use this parameter to focus the analysis on only page transition pages using custom report filters.

---

WT.ttp

WT.ttp=1

Identifies transition target pages for SmartView. The JavaScript creates and passes this query parameter if you enabled SmartView page transition tracking in the JavaScript tag and you tagged the previously viewed page with the SmartView page transition META tag. For more information, see “Configuring SmartView Using JavaScript Tags” in the SmartView User’s Guide.

Do not use this query parameter to tag your web site. You can use this parameter to focus the analysis on only page transition pages using custom report filters.

---

Stored Visitor Parameter

The Stored Visitor parameter identifies the unique visitor ID you assign to your visitors.

WT.dcsvid

WT.dcsvid=anystring

Webtrends Analytics Considerations

If you enable Visitor History in a profile, when Webtrends Analytics detects this parameter, it is stored in the Visitor History database. When the Visitor History database is exported, this parameter is exported along with each visitor for the purpose of identifying visitors. For more information about visitor history see “Visitor History Parameters” on page 24.
Visitor Data Mart Considerations

Visitor Data Mart uses this parameter to populate the External Visitor ID field of the Visitor table, which allows you to link to external visitor data in the Extended Attributes Database. For more information about the complete set of query parameters that Visitor Data Mart uses, see “Configuring Your Web Site to Collect Visitor Data Mart Data” in the Visitor Data Mart User’s Guide.

Visitor History Parameters

The Webtrends analysis process can generate and maintain parameters that support visitor profiling when you enable Visitor History in your profiles. These visitor-related parameters are stored in a Visitor History Table for each profile that has Visitor History enabled. Because Visitor History parameters are handled by Webtrends, you should not use these parameters in META tags on your web pages.

Parameters that describe elapsed time periods in days are calculated as a complete 24-hour period. Thus, if a visitor visits for the first time at 1:00 on Monday, then any visit before 1:00 on Tuesday is considered as zero days since the first visit (WT.vr.fvd), even though the actual day is different.

Most Recent Campaign Parameters

Webtrends generates and maintains these parameters when you enable Visitor History and select the Campaign History category in your profile. For more information, see “Visitor History Parameters”.

WT.vr.rac
WT.vr.rac=MostRecentCampaign
Identifies the visitor’s most recent campaign. Of all of the campaigns, the “most recent” campaign is the one that was most recently added to the visitor history table. This is a single value (no multiples allowed). This parameter is not set if the visitor has never had a campaign.

Unique Visitors for Campaigns

You can use the following set of parameters to track unique visitors for daily, weekly, monthly, quarterly, yearly, and lifetime campaigns. The active campaign list is used to determine the values. Therefore, you can get an additional campaign unique visitor when it is referenced again. These parameters are only available for reporting if you use the default campaign translation file. For more information, see “Lookup Tables for Drilldowns” in the Administration User’s Guide.

WT.vr.rac_dc
WT.vr.rac_dc=CampaignDemandChannel
Identifies the campaign demand channel. It is set to the description corresponding to the ID in the translation file.

WT.vr.rac_de
WT.vr.rac_de=CampaignDescription
This parameter is set to the description corresponding to the ID in the translation file.

WT.vr.rac_cr
WT.vr.rac_cr=CampaignCreative
This parameter is set to the creative corresponding to the ID in the translation file. A campaign creative is an attribute of a specific offer, for example, a “Buy Now!” graphic. A specific offer may consist of many creatives.
Webtrends Query Parameter Reference

Visitor History Parameters

**WT.vr.rac_ct**

WT.vr.rac_ct=CampaignCreativeType

This parameter is set to the creative type corresponding to the ID in the translation file.

**WT.vr.rac_ma**

WT.vr.rac_ma=CampaignMarketingActivity

This parameter is set to the marketing activity corresponding to the ID in the translation file.

**WT.vr.rac_mp**

WT.vr.rac_mp=CampaignMarketingProgram

This parameter is set to the marketing program corresponding to the ID in the translation file.

**WT.vr.rac_of**

WT.vr.rac_of=CampaignOffer

This parameter is set to the offer corresponding to the ID in the translation file.

**WT.vr.rac_pa**

WT.vr.rac_pa=CampaignPartner

This parameter is set to the partner corresponding to the ID in the translation file.

**WT.vr.rac_pl**

WT.vr.rac_pl=CampaignPlacement

This parameter is set to the placement corresponding to the ID in the translation file.

**Most Recent Campaign Visitors**

These parameters are generated and maintained by the Webtrends analysis process when you enable Visitor History and select the Campaign History category in your profile. For more information, see “Visitor History Parameters” on page 24.

**WT.vr.rac_d**

WT.vr.rac_d=1

Identifies a visitor’s first visit for a day for the campaign specified in WT.mc_id (campaign ID) on the hit.

**WT.vr.rac_w**

WT.vr.rac_w=1

Identifies a visitor’s first visit for a week for the campaign specified in WT.mc_id (campaign ID) on the hit.

**WT.vr.rac_m**

WT.vr.rac_m=1

Identifies a visitor’s first visit for a month for the campaign specified in WT.mc_id (campaign ID) on the hit.

**WT.vr.rac_q**

WT.vr.rac_q=1

Identifies a visitor’s first visit for a quarter for the campaign specified in WT.mc_id (campaign ID) on the hit.

**WT.vr.rac_y**

WT.vr.rac_y=1

Identifies a visitor’s first visit for a year for the campaign specified in WT.mc_id (campaign ID) on the hit.
Visitor History Parameters

Visitor "Initial" Parameters

The following list describes several visitor parameters that keep track of the “first” aspects of a visitor’s history with the site. Note that although the parameters use the terminology “first,” all preconfigured objects based on these parameters use the term “initial” (for example, Initial Referrer is the dimension based on WT.vr.fr). Webtrends generates these parameters when you enable Visitor History. For more information, see “Visitor History Parameters” on page 24.

WT.vr.ac_f

Identifies a visitor’s first visit for the campaign specified in WT.mc_id (campaign ID) on the hit.

WT.vr.fr

Identifies the visitor’s first recorded referrer. The format is the same as that for the Referring Domains dimension (for example, google.com). This is set on the first hit of the first visit and does not change afterwards.

WT.vr.fc

Identifies the visitor’s first recorded marketing campaign. The format is the same as that for the Campaign dimension. Only campaigns identified using the WT.mc_id parameter are counted. Campaigns defined solely through Webtrends Administration are not used.

This parameter is not provided until the visitor visits with a campaign. At that point the value and parameter are set and will never change.

WT.vr.fe

Identifies the visitor’s first recorded page view. The format is a page URL without query parameters. This is set on the first hit of the first visit and never changes afterwards.

Elapsed Time Parameters

These parameters are generated and maintained by the Webtrends analysis process when you enable Visitor History. For more information, see “Visitor History Parameters” on page 24.

WT.vr.fvd

Identifies the days since the visitor’s first visit. This is an integer containing the days since the visitor’s first visit. The value is truncated (for example, if 47 hours has passed since the first visit, the value is 1). This parameter is not provided on a visitor’s first visit.

When using this as a measure for the Visitor dimension, use the maximum value.

When using this as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

WT.vr.pvd

Days since the Visitor’s Previous Visit. This is an integer containing the days since the visitor’s previous visit.
This parameter makes most sense when used as a visit filter. This parameter is not provided on a visitor’s first visit.

When using this parameter as a measure for the Visitor dimension, use the maximum value. When using this parameter as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

**WT.vr.pvdb**

`WT.vr.pvdb=High/Moderate/Some/Low descriptor`

Classifies days since the visitor’s previous visit into one of four categories. This parameter is non-numeric and is used as a dimension.

You can change the number of categories by editing the `vrbucket.ini` file. By default this file is configured as follows:

```
[PVDBValues]
High = 4
Moderate = 8
Some = 12
```

The following table describes the meaning of the default values.

<table>
<thead>
<tr>
<th>[PVDBValues]</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>High=4</td>
<td>$x \leq 4$</td>
</tr>
<tr>
<td>Moderate=8</td>
<td>$4 &lt; x \leq 8$</td>
</tr>
<tr>
<td>Some=12</td>
<td>$8 &lt; x \leq 12$</td>
</tr>
</tbody>
</table>

By default, the maximum number of days for `Some` is set to 12. Any visitor whose last visit was more than 12 days ago is assigned to the Low Recency category. Recency is to the number of days since a visitor’s most recent visit.

**Example 1:** Adjust all of the ranges in `vrbucket.ini` to:

```
[PVDBValues]
High = 7
Moderate = 14
Some = 21
```

**Example 2:** To create two buckets (for example, `High/Low`), change `vrbucket.ini` to:

```
[PVDBValues]
High = 4
Moderate = 4
Some = 4
```

The result of Example 2 is:

- **High**: $x \leq 4$
- **Low**: $x > 4$

**WT.vr.ppd**

`WT.vr.ppd=DaysSincePreviousPurchase`
Identifies the days since the visitor’s previous purchase. This is an integer value.

This parameter is typically used as a visit filter. This parameter is not provided until the visitor makes the first purchase and does change on every hit that a purchase is made after that.

When using this parameter as a measure for the Visitor dimension, use the maximum value. When using this parameter as a measure for dimensions other than Visitor, you will usually configure the average value.

The sum of this measure has no meaning.

\[ WT.vr.lat \]
\[ WT.vr.lat = \text{Visit Latency} \]

Visit Latency. The visit latency is the number of days since the visitor's first visit (\( WT.vr.fvd \)) divided by the number of visit intervals (\( WT.vr.vc \)). It gives an indication of the average elapsed time between visits. This parameter is not provided on the first visit.

When using this parameter as a measure for the Visitor dimension, use the maximum value. When using this as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

**Historical Counts Parameter**

Webtrends generates and maintains these parameters by the Webtrends analysis process when you enable Visitor History. For more information, see “Visitor History Parameters” on page 24.

\[ WT.vr(vc) \]
\[ WT.vr.vc = \text{Visit Count} \]

Identifies the total number of visits recorded for a visitor. This is an integer representing the number of visits since the visitor’s first visit. When using this as a measure for the Visitor dimension, use the maximum value. When using this as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

\[ WT.vr.vcb \]
\[ WT.vr.vcb = \text{High/Moderate/Some/Low descriptor} \]

Classifies the value of the \( WT.vr.vc \) parameter into one of four categories.

This parameter is non-numeric and used as a dimension. You can change the number of categories by editing the \( vrbucket.ini \) file. By default, the file is configured as follows:

```ini
[VCBValues]
High = 25
Moderate = 15
Some = 5
```

The following table shows the meaning of the default values:

<table>
<thead>
<tr>
<th>[VCBValues]</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>High=25</td>
<td>( 25 &lt; x )</td>
</tr>
<tr>
<td>Moderate=15</td>
<td>( 15 &gt; x \leq 25 )</td>
</tr>
</tbody>
</table>
By default, the minimum value for Some is 5. Any visitor whose visit count value is less than 5 is assigned to the Low value category.

Example 1: Adjust all of the ranges in vrbucket.ini to:

```
[VCBValues]
High = 50
Moderate = 30
Some = 10
```

Example 2: To create two buckets (for example, High/Low), change vrbucket.ini to:

```
[VCBValues]
High = 25
Moderate = 25
Some = 25
```

The result of example 2 is:

- High is >25
- Low is 0 ≤ 25

**Historical Transactions/Purchases Parameters**

Webtrends generates and maintains these parameters by the Webtrends analysis process when you enable Visitor History. For more information, see “Visitor History Parameters” on page 24.

The following parameters are calculated using the transaction parameters. For more information, see “Transaction Parameters” on page 16.

**WT.vr.vv**  
**WT.vr.vv=VisitorValue**

Webtrends generates this Visitor History parameter to track the visitor’s overall value, which is the value of all purchases recorded for a visitor over time. This is a floating-point value containing the amount of money spent by this visitor back to and including the visitor’s first visit.

When using this as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

**WT.vr.ltb**  
**WT.vr.ltb=High/Moderate/Some/Low descriptor**

Webtrends generates this Visitor History parameter to classify the WT.vr.vv parameter value in one of four categories.

The value is non-numeric and is used as a custom report dimension.

You can change the number of categories by editing the vrbucket.ini file. By default, the file is configured as follows:

```
[LTBValues]
High = 750
Moderate = 500
```
Visitor History Parameters

Some $= 250$

The following table shows the meaning of the default values:

<table>
<thead>
<tr>
<th>[LTBValues]</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>High=750</td>
<td>$x &gt; 750$</td>
</tr>
<tr>
<td>Moderate=500</td>
<td>$500 &gt; x \leq 750$</td>
</tr>
<tr>
<td>Some=5</td>
<td>$250 &gt; x \leq 500$</td>
</tr>
</tbody>
</table>

By default, the minimum value for Some is 250. Any visitor whose value is less than 250 is assigned to the Low category.

Example 1: Adjust all of the ranges in `vrbucket.ini` to:

```
[LTBValues]
High = 1500
Moderate = 1000
Some = 500
```

Example 2: To create two buckets (for example, High/Low), change `vrbucket.ini` to:

```
[LTBValues]
High = 750
Moderate = 750
Some = 750
```

The result of example 2 is:

- **High** is $> 750$
- **Low** is $> 0 \leq 750$

**WT.vr.ppv**

`WT.vr.ppv=PreviousPurchaseValue`

Webtrends generates this Visitor History parameter to track the visitor’s previous purchase amount. This is a floating-point value containing the amount spent on the most recent purchase. This parameter is not generated until a visitor makes the first purchase.

When using this parameter as a custom report measure for the Visitor dimension, use the maximum value. When using this parameter as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

**WT.vr.vp**

`WT.vr.vp=VisitorPurchases`

Webtrends generates this Visitor History parameter to track the total number of purchases ever made by a visitor. The value is an integer containing the number of purchase transactions (not the total number of units purchased) back to and including the visitor’s first visit.

When using this parameter as a custom report measure for the Visitor dimension, use the maximum value. When using this parameter as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.
WT.vr.fpd
WT.vr.fpd=DaysSinceFirstPurchase
Webtrends generates this Visitor History parameter to track the days since the visitor’s first purchase. The value is an integer containing the days since the visitor’s first purchase.

This parameter is best used as a visit filter or as a custom report measure. This parameter is not generated until a visitor makes the first purchase.

When using this parameter as a measure for the Visitor dimension, use the maximum value. When using this parameter as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

WT.vr.bpd
WT.vr.bpd=DaysBeforeFirstPurchase
Webtrends generates this Visitor History parameter to track the days before the visitor’s first purchase. The value is an integer containing the number of days between the visitor’s first visit and the first purchase.

This parameter is best used as a visit filter or as a measure. This parameter is not provided until a visitor makes the first purchase.

When using this as a measure for dimensions other than Visitor, you usually configure the average value.

The sum of this measure has no meaning.

Webtrends generates the following set of parameters to track unique buyers and buyer’s status for daily, weekly, monthly, quarterly, yearly, and lifetime periods.

WT.vr_brws
WT.vr_brws=Buyer/Non-Buyer
Webtrends generates this query parameter on a visitor’s first visit of the day. Its value indicates whether a visitor has purchased in the past. Buyer indicates that the visitor has purchased before. Non-buyer indicates that the visitor has not purchased.

A visitor is considered a buyer if the WT.tx_s parameter is passed in the query string for the visit. A visitor is considered a non-buyer if WT.tx_s is not passed in the query string. In addition, Webtrends uses the following invoice parameters to evaluate whether WT.tx_s should be used for buyer determination:

- WT.tx_id and WT.tx_it
- WT.tx_i

If both WT.tx_id and WT.tx_it parameters are passed during the visit and are properly formatted, Webtrends uses them to evaluate whether WT.tx_s should be used for buyer determination. If the date and time specified by these parameters is older than the time of the hit by more than the configured invoice age limit, WT.tx_s is not used and the visitor is considered a non-buyer. The invoice age limit is set at two days, meaning that any invoices three days or older than the first hit associated with the invoice are not used for buyer determination.

If the WT.tx_i parameter is passed during the visit, Webtrends uses it to evaluate whether WT.tx_s should be used for buyer determination. Webtrends looks for the invoice number passed in for this query parameter in the visitor’s invoice history. If it has seen this invoice number before, WT.tx_s is not used and the visitor is considered a non-buyer. Three invoices are kept per visitor. Invoices more than two days old are purged from the visitor’s invoice history.
Visitor History Parameters

WT.vr.by
WT.vr.by=New Buyer/Repeat Buyer

Webtrends generates this Visitor History parameter when a visitor makes a purchase. It indicates whether the visitor is purchasing for the first time or has purchased before.

WT.vr.bt_d
WT.vr.bt_d=1

Webtrends generates this Visitor History parameter to identify the visitor’s first daily purchase. This parameter is generated and set to 1 for the first purchase from a visitor for a day.

WT.vr.bt_w
WT.vr.bt_w=1

Webtrends generates this Visitor History parameter to identify the visitor’s first weekly purchase. This parameter is generated and set to 1 for the first purchase from a visitor during a week.

WT.vr.bt_m
WT.vr.bt_m=1

Webtrends generates this Visitor History parameter to identify the visitor’s first monthly purchase. This parameter is generated and set to 1 for the first purchase from a visitor during a month.

WT.vr.bt_q
WT.vr.bt_q=1

Webtrends generates this Visitor History parameter to identify the visitor’s first quarterly purchase. This parameter is generated and set to 1 for the first purchase from a visitor during a quarter.

WT.vr.bt_y
WT.vr.bt_y=1

Webtrends generates this Visitor History parameter to identify the visitor’s first yearly purchase. This parameter is generated and set to 1 for the first purchase from a visitor during a year.

WT.vr.bt_f
WT.vr.bt_f=1

Webtrends generates this Visitor History parameter to identify the visitor’s first purchase. This parameter is generated and set to 1 for the first purchase from a visitor.

Search Engine Parameters

The Search Engine parameters keep track of initial and most recent search engines and search engine phrases for a visitor. These parameters work in conjunction with the WT.srch parameter to determine whether the referring search engine was from a paid search phrase. Webtrends generates and maintains these parameters when you enable Visitor History and select the Search Engine History category in your profile. For more information, see “Visitor History Parameters” on page 24.

WT.vr.ise
WT.vr.ise=InitialSearchEngine

Webtrends generates this Visitor History parameter to track the visitor’s initial search engine. This parameter contains the string identifying the initial search engine for a visitor. The parameter is generated with the hit where it is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is recognized. The parameter is not provided until it has been recognized and set to the initial value.

WT.vr.isep
WT.vr.isep=InitialSearchEnginePhrase
Webtrends generates this Visitor History parameter to track the visitor’s initial search engine phrase. This parameter contains the string identifying the initial search engine phrase for a visitor. The parameter is generated with the hit where the search engine is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is recognized. The parameter is not provided until it has been recognized and set to the initial value.

**WT.vr.ipd.se**

*WT.vr.ipd.se=InitialPaidSearchEngine*

Webtrends generates this Visitor History parameter to track the visitor’s initial paid search engine. This parameter contains the string identifying the initial paid search engine for a visitor. A paid search engine referrer is identified by a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.ipd.se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is recognized. The parameter is not provided until it has been recognized and set to the initial value.

**WT.vr.ipd.sep**

*WT.vr.ipd.sep=InitialPaidSearchEnginePhrase*

Webtrends generates this Visitor History parameter to track the visitor’s initial paid search engine phrase. This parameter contains the string identifying the initial paid search engine phrase for a visitor. A paid search engine referrer/phrase is identified by a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.ipd.sep` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is first recognized. The parameter is not provided until it has been recognized and set to the initial value.

**WT.vr.iog.se**

*WT.vr.iog.se=InitialOrganicSearchEngine*

Webtrends generates this Visitor History parameter to track the visitor’s initial organic search engine. This parameter contains the string identifying the initial organic search engine for a visitor. An organic search engine referrer is identified by the lack of a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.iog.se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is recognized. The parameter is not provided until it has been recognized and set to the initial value.

**WT.vr.iog.sep**

*WT.vr.iog.sep=InitialOrganicSearchEnginePhrase*

Webtrends generates this Visitor History parameter to track the visitor’s initial organic search engine phrase. This parameter contains the string identifying the initial paid search engine phrase for a visitor. An organic search engine referrer/phrase is identified by the lack of a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.iog.sep` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter never changes and is provided with the first hit of every visit after the visit in which it is first recognized. The parameter is not provided until it has been recognized and set to the initial value.

**WT.vr.r.se**

*WT.vr.r.se=MostRecentSearchEngine*
Webtrends generates this Visitor History parameter to track the visitor’s most recent search engine. This parameter contains the string identifying the most recent search engine for a visitor. The `WT.vr.r_se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter changes whenever a new search engine is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.

`WT.vr.r_se
WT.vr.r_se=Most Recent Search Engine Phrase`

Webtrends generates this Visitor History parameter to track the visitor’s most recent search engine phrase. This parameter contains the string identifying the most recent search engine phrase for a visitor. The `WT.vr.r_sep` parameter is provided with the hit where `WT.vr.r_se` is recognized as the referring site. The value of the parameter changes whenever a new search engine/search engine phrase is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.

`WT.vr.rpd_se
WT.vr.rpd_se=Most Recent Paid Search Engine`

Webtrends generates this Visitor History parameter to track the visitor’s most recent paid search engine. This parameter contains the string identifying the most recent paid search engine for a visitor. A paid search engine referrer/phrase is identified by a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.rpd_se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter changes whenever a new search engine is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.

`WT.vr.rpd_sep
WT.vr.rpd_sep=Most Recent Paid Search Engine Phrase`

Webtrends generates this Visitor History parameter to track the visitor’s most recent paid search engine phrase. This parameter contains the string identifying the most recent paid search engine phrase for a visitor. A paid search engine referrer/phrase is identified by a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.rpd_se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter changes whenever a new search engine is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.

`WT.vr.rog_se
WT.vr.rog_se=Most Recent Organic Search Engine`

Webtrends generates this Visitor History parameter to track the visitor’s most recent organic search engine. This parameter contains the string identifying the most recent organic search engine for a visitor. An organic search engine referrer/phrase is identified by the lack of a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr.rog_se` parameter is provided with the hit where it is recognized as the referring site. The value of the parameter changes whenever a new search engine/organic search engine phrase is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.
**Visitor History Parameters**

Visitor Tracking parameters let you track daily, weekly, monthly, quarterly, yearly, and all-time unique visitors. Webtrends generates and maintains these parameters when you enable Visitor History and select the Visit History category in your profile. For more information, see “Visitor History Parameters” on page 24.

**Visitor Tracking Parameters**

**WT.vr rog sep**

**WT.vr rog sep=Most Recent Organic Search Engine Phrase**

Webtrends generates this Visitor History parameter to track the visitor’s most recent organic search engine phrase. This parameter contains the string identifying the most recent organic search engine phrase for a visitor. An organic search engine referrer/phrase is identified by the lack of a `WT.srch=1` parameter in the query field of hit query string. The `WT.vr rog sep` parameter is provided with the hit where `WT.vr rog sep` is recognized as the referring site. The value of the parameter changes whenever a new search engine/search engine phrase is recognized. It is provided with the first hit of every visit after the visit in which it is recognized. It changes whenever a new search engine is recognized. The parameter is not provided until it has been recognized and set to a first value.

**WT.vr vt_d**

**WT.vr vt_d=1**

Webtrends generates this Visitor History parameter to track daily visitor daily activity. This parameter is present and set to 1 on a new visitor’s first hit for the day.

**WT.vr vt_w**

**WT.vr vt_w=1**

Webtrends generates this Visitor History parameter to track weekly visitor activity. This parameter is present and set to 1 on a new visitor’s first hit for the week.

**WT.vr vt_m**

**WT.vr vt_m=1**

Webtrends generates this Visitor History parameter to track monthly visitor activity. This parameter is present and set to 1 on a new visitor’s first hit for the month.

**WT.vr vt_q**

**WT.vr vt_q=1**

Webtrends generates this Visitor History parameter to track quarterly visitor activity. This parameter is present and set to 1 on the first hit from a new visitor during a quarter.

**WT.vr vt_y**

**WT.vr vt_y=1**

Webtrends generates this Visitor History parameter to track yearly visitor activity. This parameter is present and set to 1 on the first hit from a new visitor during a year.

**WT.vr vt_f**

**WT.vr vt_f=1**

Webtrends generates this Visitor History parameter to track a visitor’s first hit. This parameter is present and set to 1 on the first hit from a new visitor.

**WT.vr piv_d**

**WT.vr piv_d=1**

Webtrends generates this Visitor History parameter to track daily page of interest activity. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest during a day.
Visitor History Parameters

- **WT.vr.piv_w**
  - WT.vr.piv_w=1
  - Webtrends generates this Visitor History parameter to track weekly page of interest activity. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest during a week.

- **WT.vr.piv_m**
  - WT.vr.piv_m=1
  - Webtrends generates this Visitor History parameter to track monthly page of interest activity. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest during a month.

- **WT.vr.piv_q**
  - WT.vr.piv_q=1
  - Webtrends generates this Visitor History parameter to track quarterly page of interest activity. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest during a quarter.

- **WT.vr.piv_y**
  - WT.vr.piv_y=1
  - Webtrends generates this Visitor History parameter to track yearly page of interest activity. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest during a year.

- **WT.vr.piv_f**
  - WT.vr.piv_f=1
  - Webtrends generates this Visitor History parameter to track a visitor’s first page of interest. This parameter is present and set to 1 on the first hit from a new visitor to a page of interest.

- **WT.vr.cgv_d**
  - WT.vr.cgv_d=1;...
  - Webtrends generates this Visitor History parameter to track daily content group activity. This parameter is present and set to 1 for the first hit from a new visitor to a content group during a day. If multiple content groups are specified on the hit, this parameter contains as many values as there are content groups.

- **WT.vr.cgv_w**
  - WT.vr.cgv_w=1;...
  - Webtrends generates this Visitor History parameter to track weekly content group activity. This parameter is present and set to 1 for the first hit from a new visitor to a content group during a week. If multiple content groups are specified on the hit, this parameter contains as many values as there are content groups.

- **WT.vr.cgv_m**
  - WT.vr.cgv_m=1;...
  - Webtrends generates this Visitor History parameter to track weekly content group activity. This parameter is present and set to 1 for the first hit from a new visitor to a content group during a month. If multiple content groups are specified on the hit, this parameter contains as many values as there are content groups.

- **WT.vr.cgv_q**
  - WT.vr.cgv_q=1;...
  - Webtrends generates this Visitor History parameter to track quarterly content group activity. This parameter is present and set to 1 for the first hit from a new visitor to a content group during a quarter.

- **WT.vr.cgv_y**
  - WT.vr.cgv_y=1;...
Webtrends generates this Visitor History parameter to track yearly content group activity. This parameter is present and set to 1 for the first hit from a new visitor to a content group during a year. If multiple content groups are specified on the hit, this parameter contains as many values as there are content groups.

\[ \text{WT.vr.cgv}_f \]
\[ \text{WT.vr.cgv}_f=1;... \]

Webtrends generates this Visitor History parameter to track a visitor’s first hit to a content group. This parameter is present and set to 1 for the first hit from a new visitor to a content group. If multiple content groups are specified on the hit, this parameter contains as many values as there are content groups.

Visitor Segmentation Parameters

Visitor Segmentation parameters allow you to store the most recent value of a segmentation query parameter for inclusion in your reports. For example, your travel web site tracks visitors using a segmentation parameter, such as \( \text{WT.seg}_1 \), to identify the visitor’s “traveler type.” The result of the visitor’s most recent traveler type value is stored in the \( \text{WT.vhseg}_1 \) parameter in the Visitor History database. Custom reports you create that use the \( \text{WT.vhseg}_1 \) parameter as a dimension show statistics for the most recent value of the key parameter.

The most recent value of the key parameter, \( \text{WT.seg}_x \) is stored in the corresponding result parameter. To report on visitor segmentation data, create a custom report that uses your result parameter as a dimension.

Webtrends generates and maintains these parameters when you enable Visitor History and select the Custom Visitor Segmentation category in your profile. You must also implement the \( \text{WT.seg} \) parameter on your web pages. For more information about \( \text{WT.seg} \), see “Segment Parameter” on page 19.

\[ \text{WT.vhseg}_1 \]
\[ \text{WT.vhseg}_1=\text{VisitorSegment1Result} \]
\[ \text{WT.vhseg}_2 \]
\[ \text{WT.vhseg}_2=\text{VisitorSegment2Result} \]
\[ \text{WT.vhseg}_3 \]
\[ \text{WT.vhseg}_3=\text{VisitorSegment3Result} \]
\[ \text{WT.vhseg}_4 \]
\[ \text{WT.vhseg}_4=\text{VisitorSegment4Result} \]

SDC-Generated Visitor Parameters

The following subsections discuss visitor-related parameters that are generated and maintained by SmartSource Data Collector (SDC).

Visitor Tracking Parameters

Visitor Tracking parameters allow you to track daily, weekly, monthly, quarterly, and yearly unique visitors. SDC inserts these parameters into the cs-uri-query strings.

\[ \text{WT.vt_tlv} \]
\[ \text{WT.vt_tlv}=\text{UNIX Time} \]
SDC generates this parameter to identify the time of the visitor’s last visit. The value is expressed as the number of seconds since 1970 (standard UNIX time), which is calculated using information stored in the third-party cookie value. SDC only sets this parameter at the start of a new visit. On a visitor’s first visit, the value is set to zero. If you disable cookie tracking, this parameter is not generated or passed in the query string.

Visitor Data Mart uses this query parameter to determine whether a new visit should also be counted as a new daily, weekly, monthly, quarterly, or yearly visitor. For example, if the day of the new visit is different than the day of the previous visit, relative to the GMT offset of the JavaScript tag, the visit is counted as a new daily visit.

\[ \text{WT.vt}_f_tlv \]
\[ \text{WT.vt.f.tlv = UNIX Time} \]

SDC generates this parameter to identify the time of the visitor’s last visit. The value is expressed as the number of seconds since 1970 (standard UNIX time), which is calculated using information stored in the first-party cookie value. SDC only sets this parameter at the start of a new visit. On a visitor’s first visit, the value is set to zero. If you disable first-party cookie tracking in the JavaScript tag, this parameter is not generated or passed in the query string.

Visitor Data Mart uses this query parameter to determine whether a new visit should also be counted as a new daily, weekly, monthly, quarterly, or yearly visitor. For example, if the day of the new visit is different than the day of the previous visit, relative to the GMT offset of the JavaScript tag, the visit is counted as a new daily visit.

\[ \text{WT.vt}_d \]
\[ \text{WT.vt.d = 1} \]

SDC generates this parameter to track daily visitors for Express Analysis. This parameter is generated and set to 1 for the first hit from a new visitor for a day.

\[ \text{WT.vt.a.d} \]
\[ \text{WT.vt.d = 1} \]

SDC generates this parameter to track daily visitors for Account Rollup Data Sources. This parameter is generated and set to 1 for the first hit from a new visitor for a day for a given account. This parameter is used in conjunction with Account Rollup Profiles.

\[ \text{WT.vt.f.d} \]
\[ \text{WT.vt.f.d = 1} \]

SDC generates this parameter to track daily visitors for Express Analysis only. This parameter is generated and set to 1 for the first hit from a new visitor when you configure your SmartSource Data Source to use the First-Party Cookie JavaScript. The First-Party Cookie JavaScript generates this parameter and its value.

\[ \text{WT.vt.s} \]
\[ \text{WT.vt.s = 1} \]

SDC generates this parameter to track visitor sessions for Express Analysis and Visitor Data Mart. This parameter is generated and set to 1 for the first hit for a new session. Cookie tracking must be enabled to set this query parameter.

\[ \text{WT.vt.a.s} \]
\[ \text{WT.vt.a.s = 1} \]

SDC generates this parameter to track visitor sessions for Account Rollup Data Sources. Applies to Real Time analysis only. This parameter is generated and set to 1 for the first hit for a new session for a given account. This parameter is used in conjunction with Account Rollup Profiles.
**WT.vt_f_s**

`WT.vt_f_s = 1`

SDC generates this parameter to track new visitor sessions for Express Analysis and Webtrends Visitor Data Mart.

This parameter is generated and set to 1 for the first hit for a new session. The First-Party Cookie JavaScript generates this parameter.

**WT.vt_f**

`WT.vt_f = 1`

SDC generates this parameter to track new and returning visitors. This parameter is generated and set to 1 for the first hit from a new visitor. This parameter is set to 2 if the visitor’s browser does not accept cookies. The First-Party Cookie JavaScript generates this parameter.

**WT.vt_f_a**

`WT.vt_f_a = 1`

SDC generates this parameter to track new visitors for an account. This parameter is generated and set to 1 for the first hit from a new visitor for a given account. This parameter is set to 2 if the visitor’s browser does not accept cookies. The First-Party Cookie JavaScript generates this parameter. This parameter is used in conjunction with Account Rollup Profiles.

**WT.vt_sid**

`WT.vt_sid = identifier`

SDC generates this parameter to identify visitor sessions for Visitor Data Mart. This parameter is generated on every hit to identify the visitor session.

This identifier is formed by concatenating two pieces of data:

- **Identifier:** The value of `WT.co_f`. If `WT.co_f` is not present on the incoming hit, SDC generates this value. This is the unique identifier that is generated at the time of a new visit. It is 32-character hexadecimal number (0-9, a-f)

- **Timestamp:** The time when the session began. This is the number of milliseconds since January 1, 1970.

---

**Note**

The First Party Cookie Javascript no longer emits `WT.vt_sid`. Rather, it now emits the two components (Identifier and timestamp) as separate parameters. See `WT.vtid` and `WT.vtvs` for more information.

---

**Example:**

`WT.vt_sid = 10.61.19.29:3899933120:29768655.1141069`

**WT.vtid**

`WT.vtid = identifier`

This parameter is used to identify visitors for Visitor Data Mart. By default, this identifier is provided by Webtrends Data Collection Servers using an in-line JavaScript request for `wtid.js` at the time of a new visit. Alternatively, this parameter value can come from a custom query parameter or customer cookie.

Identifier: Any alphanumeric string that uniquely identifies a visitor. By default this value is the same as the `WT.co_f` parameter.
WT.vtvs
WT.vtvs=timestamp

The JavaScript tag generates this parameter to identify visitor sessions for Webtrends Visitor Data Mart. This parameter is generated on every hit to identify the visitor session.

Timestamp: The time when the session began. This is the number of milliseconds since January 1, 1970.

Cookie Detection Parameters

This set of parameters allows Webtrends Analytics to tie a visitor’s first hit with the rest of the visitor session. These parameters are generated and maintained by SmartSource Data Collector (SDC).

WT.co
WT.co=Yes/No

SDC generates this parameter to determine whether the visitor’s browser supports and is configured to accept cookies. Valid values are Yes and No.

Example:
var coQueryParam = "&WT.co=" + navigator.cookieEnabled() ? "Yes" : "No";

WT.co_d
WT.co_d=Cookie_Data

SDC generates this parameter the first time it attempts to set the cookie. The value is set to the value of the cookie. This parameter is generated only for “first visit” hits.

Webtrends Analytics and Visitor Data Mart use this parameter for “session stitching.” The first hit of the first visitor session (which may not have a cookie) gets the cookie value in the WT.co_d parameter. Subsequent hits that have the same value for the Webtrends cookie can be tied together with the WT.co_d hit to form a complete picture of the session. Also, SDC passes the WT.co_d value; it is not passed from the visitor’s web browser.

The Webtrends cookie format contains the IP address of the cookie’s connection address and the creation time. The creation time is represented as the number of seconds and nanoseconds since 1970 (standard UNIX time). A checksum is appended to the cookie value. The following example shows the format of the cookie value:

WEBTRENDS_ID=IP Address:SSSSSSSSS.NNNNNN:checksum

The following example shows the WT.co_d parameter with a Webtrends cookie value:

WT.co_d=192.168.100.40:1045156016.29542554::A2D3FC34517CE562A9D4E33EF85D7B7F

WT.co_a
WT.co_a=Cookie_Data

SDC generates this parameter the first time it attempts to set the account cookie. Note that this parameter is generated only for hits on the first visit. This parameter is used to track visitor sessions across multiple accounts in Webtrends Analytics On Demand or multiple SmartSource data sources in Webtrends Analytics software.

This parameter is generated and the value is set to the account rollup cookie’s value if SDC attempted to set a first-time cookie for a given account. The global rollup cookie is named ACookie. The global rollup cookie contains an encoded account rollup cookie named WT_ACCT. It contains the IP address of the cookie’s connection address and the creation time. The creation time is represented as the number of seconds and nanoseconds since 1970 (standard UNIX time). A checksum is appended to the cookie value. The following example shows the format of the cookie value:

WT_ACCT=IP Address:SSSSSSSSS.NNNNNN:checksum
The following example shows the WT.co_a parameter with an account rollup cookie:

WT.co_a=192.168.100.40-1045156016.29542554

**WT.co_f**

WT.co_f=uniqueIdentifier

SDC generates this parameter when you enable first-party cookie tracking in the Webtrends JavaScript tag. This parameter is passed on every hit so that Webtrends Analytics and Visitor Data Mart can use it for visitor session tracking.

**WT.co_f** uses the following format:

*IP Address-UniqueIdentifier:LastVisitTime.VisitStartTime*

Where the unique identifier always 9-characters long and times use the epoch timestamp.

Example:

WT.co_f= 192.168.100.40-123456789:1045156016.29542554

You can configure the method that you want to use for your first party cookies in the SmartSource Data Source settings.

**URL Truncation Parameter**

SDC uses this parameter to overcome a maximum URL length limitation imposed by Internet Explorer (Microsoft Knowledge Base Article – 208427). The URL length must be 2048 or less. If the Webtrends JavaScript tag generates a URL in excess of 2048 characters and the client browser is Microsoft Internet Explorer, the hit is truncated and is passed to SDC.

**WT.tu**

WT.tu=1

SDC generates this parameter and sets it to 1 if the URL was deemed too long and truncated by the JavaScript tag. If present, SDC writes an error and discards the hit.

You can configure the logtruncatedhits setting to log the truncated hit rather than discard it.

**HTTP Headers**

You may want to access custom HTTP request headers. These headers can be inserted by third-party products such as load balancers, application servers, or web server plug-ins. This parameter is assigned the value of the specified HTTP header, which can then be referenced in a Webtrends custom report.

**WT.hdr.HTTP Header**

WT.hdr.HTTP Header=Value

If the header is present in the incoming request, the header name is appended to **WT.hdr.** and the header value is assigned to the value. For example, suppose that a customer wants to log the Accept: header, and it comes in as Accept: */*. The resultant parameter would be WT.hdr.Accept=*/*. Note that values are URL encoded.

**JavaScript Tag Version**

**WT.tv**

WT.tv=major.minor.revision
The JavaScript tag contains this parameter, which specifies the version of the Webtrends JavaScript tag that is currently deployed. The value is passed as `major.minor.revision` where `major.minor` specifies the Webtrends Analytics version and `revision` specifies the version of the JavaScript tag.

Although this parameter is not used in reports, it can be useful for Support when troubleshooting a tagging problem.

### Site ID

The `WT.site` parameter enables the website owner to supply a site ID for one or more events when multiple events occur on the same hit. The JavaScript tag generates and sets this parameter to the domain name/Site Id for cross-domain Visitor Data Mart use, enabling users to select only events for a certain ID when querying the database.

Currently, this parameter is not supported in Tag Builder and must be configured manually.

### Traffic Source

The `WT.tsrc` parameter enables the website owner to override the default Traffic Source values with a custom value. By default, Webtrends reports on the source of traffic based on its referrer using the following values: "Paid Search," "Non-Search Campaigns," "Organic Search," "Other Referrers," or "Direct". This information is tracked in the custom Traffic Source dimension. You can use the `WT.tsrc` parameter to override any of the default Traffic Source values with a custom value.

### Event Tracking

The `WT.dl` parameter specifies the kind of event tracked. The `WT.dl` parameter passes a set of identifiers. Each identifier is associated with a given event which is typically mouse related. Identifiers are numeric, and are assigned inside an event handler. The `WT.dl` parameter can be used to filter event related traffic.

The JavaScript tag generates this parameter with the appropriate event id as shown in the following table:

<table>
<thead>
<tr>
<th>Event Id</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Page View</td>
<td>Generated when page is loaded.</td>
</tr>
<tr>
<td>20</td>
<td>Download</td>
<td>Generated when a download link is clicked. Download links are onsite links whose file type matches a configurable list of download file types.</td>
</tr>
<tr>
<td>Event Id</td>
<td>Event</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>21</td>
<td>Anchor</td>
<td>Generated when an anchor link is clicked. Anchor links are on-site links that contain anchor text.</td>
</tr>
<tr>
<td>22</td>
<td>Dynamic &quot;javascript&quot;</td>
<td>Generated when a link containing a JavaScript URL is clicked. Example: <code>&lt;a href=&quot;javascript:alert('Good Morning')&quot;&gt;Good Morning&lt;/a&gt;</code></td>
</tr>
<tr>
<td>23</td>
<td>Dynamic &quot;mailto:&quot;</td>
<td>Generated when a link containing a mailto URL is clicked. Example: <code>&lt;a href=&quot;mailto:firstname.lastname@company.com?subject=Good%20Morning&quot;&gt;Send Email&lt;/a&gt;</code></td>
</tr>
<tr>
<td>24</td>
<td>Off-site</td>
<td>Generated when an off-site link is clicked. Off-site links are page elements that lead to web sites that are not instrumented with the customers SmartSource tags. The list of on-site domains is user-configurable in Webtrends Administration.</td>
</tr>
<tr>
<td>25</td>
<td>Right-click</td>
<td>Generated when a download link is right-clicked. Download links are on-site links whose file type matches a configurable list of download file types. The list of download file types is user-configurable in Webtrends Administration.</td>
</tr>
<tr>
<td>26</td>
<td>Form Button - Get method</td>
<td>Generated when a form button is clicked. Button is enclosed inside a form, and the method is GET.</td>
</tr>
<tr>
<td>27</td>
<td>Form Button - Post method</td>
<td>Generated when a form button is clicked. Button is enclosed inside a form, and the method is POST.</td>
</tr>
<tr>
<td>28</td>
<td>Form Button - Input tag</td>
<td>Generated when a form button is clicked. Button is not enclosed inside a form, but is enclosed in an <code>&lt;input&gt;</code> tag.</td>
</tr>
<tr>
<td>29</td>
<td>Form Button - Button tag</td>
<td>Generated when a form button is clicked. Button is not enclosed inside a form, but is enclosed in a <code>&lt;button&gt;</code> tag.</td>
</tr>
<tr>
<td>30</td>
<td>Image Map</td>
<td>Generated when an image map is clicked.</td>
</tr>
</tbody>
</table>
Parent DIV/Table ID

WT.nv

WT.nv=<id||class>

This parameter contains the id or class of the parent DIV or TABLE of the element that was clicked. The WT.nv value allows you to see the areas on a page to which an element was clicked belongs.

Click-Based Tracking

WT.es

WT.es=hostname/page

This parameter permits tracking click-based events and the source page from which they originated. The JavaScript tag forms this parameter by concatenating the dcssip and dcsuri parameters to determine what page a user was on when an "event" occurred. For example, when a user clicks on a link, the event source shows what page the user was viewing when the click occurred. Although in the case of a page view, these URLs are identical, in the case of a click the dcsuri is actually the "destination URL" of the click, while the event source is the "source URL" of the click.

DCSID

WT.dcs_id

WT.dcs_id=DCSID

This parameter contains the value of the DCSID that generated the hit. The Standard Analysis Engine, Express Analysis Engine, and Event Database Loader pass the DCSID as value of for this parameter. This parameter becomes most useful when tracking multiple DCSIDs in SmartSource files from multiple sites. In this case, you can use this parameter to segment your report data by site. For example, you can use this parameter as a dimension in a custom report to report on activity for each site.

---

Note

If you look at your SmartSource files, you will not find this parameter in the query string. This is because Webtrends Analytics adds it to the query parameter set during analysis.

---

SDC-Parameter Override Parameters

You can use the parameters in this section to override SDC parameters on the client side. Consider the following example:

If you want a specific page, /xyz.html, logged to the cs-uri-stem field, you can assign the page name to the dcsuri parameter in the JavaScript tag as shown in the Modified Tag.

Default JavaScript Tag:

DCS.dcsuri="window.location.pathname;"

Modified JavaScript Tag:

DCS.dcsuri="/xyz.html;"

However, because modifying JavaScript is error-prone, you could instead use the DCS.dcsuri parameter in a META tag to override the dcsuri assignment in the JavaScript tag. Your META tag would look like this:

---

Spring 2011 Release
Keep in mind that because these parameters simply override assignments in the JavaScript tag, the parameters themselves are not actually sent to SDC. The JavaScript tag contains a custom object named DCS. This object contains property name/value pairs that are used to form query parameters sent to SDC. To continue our example, the JavaScript tag first extracts the META tag information and performs the following assignment:

```javascript
DCS.dcsuri = '/xyz.html'
```

Next, the JavaScript tag iterates through all name/value pairs in the DCS object and forms query parameters.

In our example, the following query parameter is formed:

```javascript
&dcsuri = /xyz.html
```

Note that the custom object name itself (DCS) is not sent to SDC.

**DCS.dcsref**

```javascript
DCS.dcsref = 'Referrer'
```

This parameter is assigned to the dcsref parameter before the hit is sent to SDC. The value is included in the cs(Referer) field of the log file.

**DCS.dcssip**

```javascript
DCS.dcssip = 'Domain'
```

This parameter is assigned to the dcssip parameter before the hit is sent to SDC. The value is included in the cs-host field of the log file.

**DCS.dcsua**

```javascript
DCS.dcsua = 'user agent'
```

This parameter is assigned to the dcsua parameter before the hit is sent to SDC. Value is included in the cs(user agent) field. Use a plug sign to encode spaces rather than %20.

**DCS.dcsuri**

```javascript
DCS.dcsuri = 'uri-stem'
```

This parameter is assigned to the dcsuri parameter before the hit is sent to SDC. The value is included in the cs-uri-stem field of the log file.

**DCS.dcspro**

```javascript
DCS.dcspro = 'Protocol'
```

This parameter is assigned to the dcspro parameter before the hit is sent. The value is included in the cs(Version) field of the log file.

**DCS.dcsqry**

```javascript
DCS.dcsqry = 'uri-query'
```

This parameter is assigned to the dcsqry parameter before the hit is sent. The value is included in the cs-uri-query field of the log file.

**DCS.dcsaut**

```javascript
DCS.dcsaut = 'authenticated username'
```

This parameter is assigned to the dcsaut parameter before the hit is sent to SDC. The value is included in the cs-user-name field.

**DCS.dcsmet**

```javascript
DCS.dcsmet = 'method'
```
This parameter is assigned to the `dcsmet` parameter before the hit is sent to SDC. The value is included in the `cs-method` field.

**DCS.dcssta**

`DCS.dcssta=status`

This parameter is assigned to the `dcssta` parameter before the hit is sent to SDC. The value is included in the `sc-status` field.

**DCS.dcsbyt**

`DCS.dcsbyt=bytes`

This parameter is assigned to the `dcsqry` parameter before the hit is sent to SDC. The value finally is included in the `sc-bytes` field.

**DCS.dcsclp**

`DCS.dcsclp=ip address`

This parameter is assigned to the `dcsclp` parameter before the hit is sent to SDC. Value is included in the `c-ip` field.

**DCS.dcsua**

`DCS.dcsua=user agent`

This parameter is assigned to the `dcsua` parameter before the hit is sent to SDC. Value is included in the `cs(user agent)` field. Use a plug sign to encode spaces rather than `%20`.

## Conversion Plug-In Parameters

The Webtrends encoding conversion plug-in uses the parameters in this section during the character encoding conversion process. The Webtrends JavaScript tag generates these parameters when the `gi18n` global variable is set to `true` and when a web page contains DCSext query parameters.

For more information about DCSext query parameters, see “Customizing the Webtrends JavaScript Tag” in *Webtrends Analytics On Demand Implementation Guide*. For more information about the encoding conversion plug-in see “Internationalization and Webtrends” in the Administration User’s Guide.

**WT.dep**

`WT.dep=DCSext parameter1[;DCSext parameter2...]`

Contains a semicolon-delimited list of the custom DCSext query parameters on a web page. The encoding conversion plug-in uses this information to identify the parameters that are known to be encoded in UTF-8. For example, a web page that contains `DCSext.abc=655` and `DCSext.xyz=889` would be captured by the JavaScript tag as `WT.dep=abc;xyz`.

---

**Note**

Do not use this parameter for collecting data.
Mobile Web and Mobile Application Parameters

These parameters are used with the Webtrends Client Libraries for Mobile Applications to track both web and application activity associated with mobile devices. These values have a maximum length of 512 characters. For more information about using these parameters with the Mobile Libraries, visit developer.webtrends.com.

Application Ad Click

WT.a_ac
WT.a_ac=1
Indicates that an ad was clicked within a mobile application.

Application Ad Impression

WT.a_ai
WT.a_ai=1
Indicates that one or more ads was viewed in a mobile application. Typically, this parameter is used with the WT.a_an parameter to specify one or more ads viewed in a single application screen.

Application Ad Name

WT.a_an
WT.a_an=Name[;…]
Specifies one more names of ads presented in a mobile application. You can pass multiple ad names using semicolons to delimit the list. The maximum length for each Name is 64 bytes.

Application Name

WT.a_nm
WT.a_nm=application name
Specifies the name of a mobile application. This parameter can be used to report on the usage of different applications.

Application Category

WT.a_cat
WT.a_cat=application category
Specifies a category of mobile application such as Games, Health & Wellness, or Productivity. This parameter can be used to report on the usage of different types of apps.

Application Publisher

WT.a_pub
WT.a_pub=application publisher
Specifies the developer, publisher, or vendor for an application. This parameter is required.
Mobile Web and Mobile Application Parameters

Connection Type

\texttt{WT.ct=connectionType}

Specifies the connection type used to send the data. Many mobile devices go in and out of various connection states such as wifi, 3G, Edge, or 7.1G. To avoid sending a flood of this data as the connection fluctuates, the best practice is to send this parameter only once per session. The Webtrends Mobile Application Libraries use this best practice.

Country

\texttt{WT.g_co=Country Code}

Specifies the country of origin for mobile traffic using a country code.

Device Model

\texttt{WT.dm=device model}

Specifies the model of the mobile device that an application is running on, for example the model of a mobile phone. For example, \texttt{WT.dm=Motorola Droid} or \texttt{WT.dm=iPhone 3GS}.

Event Time Stamp

\texttt{WT.ets=epoch time stamp}

Specifies an epoch (unix) time stamp for an event. This time stamp is recorded so events that occur when a device is offline can be sent as an ordered batch when the device comes back online. For example, For example, \texttt{WT.ets=1006725234}.

Event Type

\texttt{WT.ev=eventType}

Specifies the type of event activity on a mobile device. For example, this parameter can be used to differentiate activities such as clicks, swipes, and content views.

Geolocation Coordinates

\texttt{WT.gc=latitude,longitude}

Specifies the latitude and longitude of a mobile device.
Document Revision History

Table 1: Document Revision History contains a summary of changes made to this document beginning with the release of Webtrends Analytics, version 8.7.

<table>
<thead>
<tr>
<th>Software Version</th>
<th>Date of Last Update</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall 2010 Release</td>
<td>November 2010</td>
<td>Added more parameters to support new Mobile Report Pack.</td>
</tr>
<tr>
<td>Spring 2010 Release</td>
<td>February 2010</td>
<td>Added “Mobile and Application Parameters.”</td>
</tr>
<tr>
<td>Fall 2009 Release</td>
<td>November, 2009</td>
<td>Corporate Branding Changes</td>
</tr>
<tr>
<td>v8.7d</td>
<td>July, 2009</td>
<td>Added footer link to Documentation Center.</td>
</tr>
<tr>
<td>v8.7</td>
<td>March, 2009</td>
<td>• Added Document Revision History section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Corrected event ID typo’s (for Dynamic “Mail To” and “Off-site” events) for “WT: dl” on page 42</td>
</tr>
</tbody>
</table>