



WebTrends Reporting Service

Customer Integration Guide

May 2003 Edition
© 2003 NetIQ Corporation

Disclaimer

NetIQ Corporation makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, NetIQ Corporation reserves the right to revise this publication and to make changes to its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, NetIQ Corporation makes no representations or warranties with respect to any NetIQ Corporation product, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, NetIQ Corporation reserves the right to make changes to any and all parts of NetIQ Corporation products at any time, without obligation to notify any person or entity of such changes.

Trademarks

AuditTrack, CommerceTrends Reporting Server, and WebTrends are registered trademarks of NetIQ Corporation.

Additional trademarks of NetIQ Corporation include: ClusterTrends, DBTrends, Enterprise Reporting Server, Enterprise Suite, Enterprise Suite for Lotus Domino, FastTrends, Log Analyzer, Firewall Suite, Security Analyzer, Virtual WebTrends, WebTrends Live, VRM, eVRM, and WebTrends Professional Suite.

Other brands and their products are trademarks or registered trademarks of their respective holders.

Product Support

If you should have any questions during the use of the WebTrends Reporting Service, please contact our Technical Support Department at:

Telephone: (503) 223-3023,

E-mail: WRS-Support@NetIQ.com.

Table of Contents

Chapter 1	
Overview	5
Interactions Between the Client Browser and WebTrends	5
Chapter 2	
Customizing Your META Tags	9
General META Tag Information	11
META Tag Descriptions	11
To track content groups	11
To track servers	12
To track marketing campaigns	13
To track profile/subprofile generation	14
To track revenue	15
To track shopping cart activity	16
To track Advertising Views	19
To track Advertising Clicks	19
To track customized URLs	20
To track page titles	23
Chapter 3	
Deployment of the META Tags	25
Copy the code to each page	25
Using Include Files	25
Server-Side Include Files	25
JavaScript Source Includes	26
Header and Footer Templates	27
Web Page Editors	27
Generating and implementing WebTrends JavaScript using a common template	27

Chapter 1

Overview

WebTrends Reporting Service uses a JavaScript code that you place on your web pages. When a visitor downloads a page from their web site, the JavaScript initiates interactions between the visitor's browser and WebTrends. The JavaScript code collects data about the visitor's browser and activity and transmits it to an analysis engine. The analysis engine aggregates this data with that of all other visitors to the your site, stores it, and makes it available to you in reports that you can view and download.

Interactions Between the Client Browser and WebTrends

After deploying the WebTrends JavaScript code on your web pages, you are set up for a series of interactions between your visitors, your web site, and WebTrends. The first interaction is between your visitors and your web site.

When a visitor downloads a page from your web site, this code initiates an interaction between the visitor's browser and transmits that data to the SmartSource Data Collector. The data is then aggregated with that of all other visitors to your site, stored, and made available to you in reports. As a registered WebTrends Reporting Service subscriber, you can log on to browse and analyze reports on their site activity.

Figure 1-1 shows an overview of the interaction process.

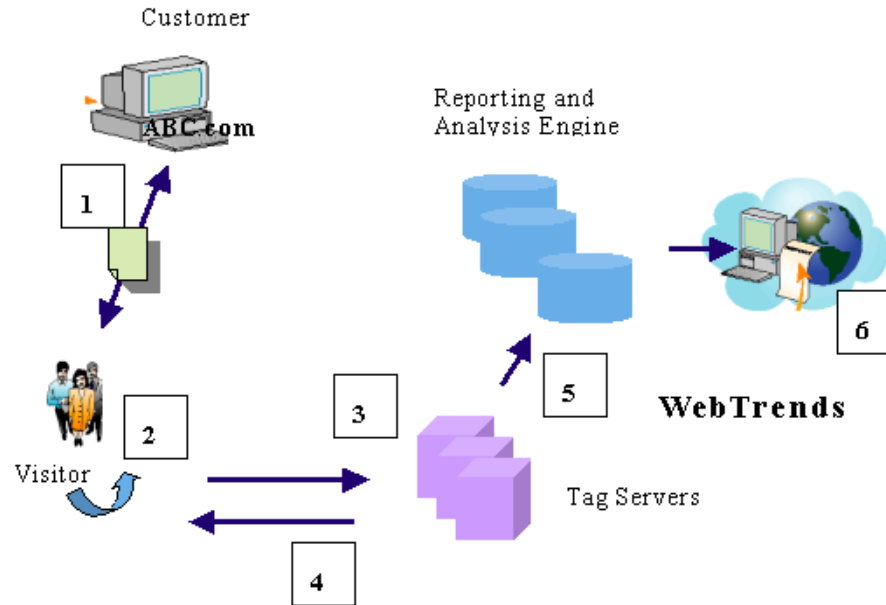


Figure 1-1 Overview of the interaction process

The interactions behave as follows:

1. You receive JavaScript code from WebTrends and place it on your web pages.
2. The JavaScript begins to execute and gathers individual pieces of information, such as URL, page title, browser type, browser version, screen resolution, eCommerce variables, and content group data.
3. The information is sent to WebTrends via the HTML information request (statse.webtrends.com/.../dcs.gif) with the information added to the query string.
4. A pool of web servers responds to the <IMG...> request by returning a 1x1 pixel GIF to the client along with cookies. The cookies contain individualized tracking information for each client.
5. The information is written to log files and analyzed by WebTrends software.

6. WebTrends send your report to you, or you can log into WebTrends and view your report.

All interactions between your visitors and WebTrends occur at the browser on the client side (your visitor's side). There is no interaction between WebTrends and your web site servers.

Chapter 2

Customizing Your META Tags

After inserting the WebTrends JavaScript, you will need to modify the META tags in your web page file, because the JavaScript interacts with the META tags and places the resulting information in the log file.

The following sections discuss all of the types of META Tags that are affected by the JavaScript. Note that the META tags don't have to appear in a particular order on your web page. However, they are presented here in an order that can help you more readily understand the structure of your web page.

WebTrends offers one of a few data tagging implementations that segregate page-specific information from the main script, maximizing code modularity and reuse.

Figure 2-1 illustrates an overview of the JavaScript/META tag interaction.

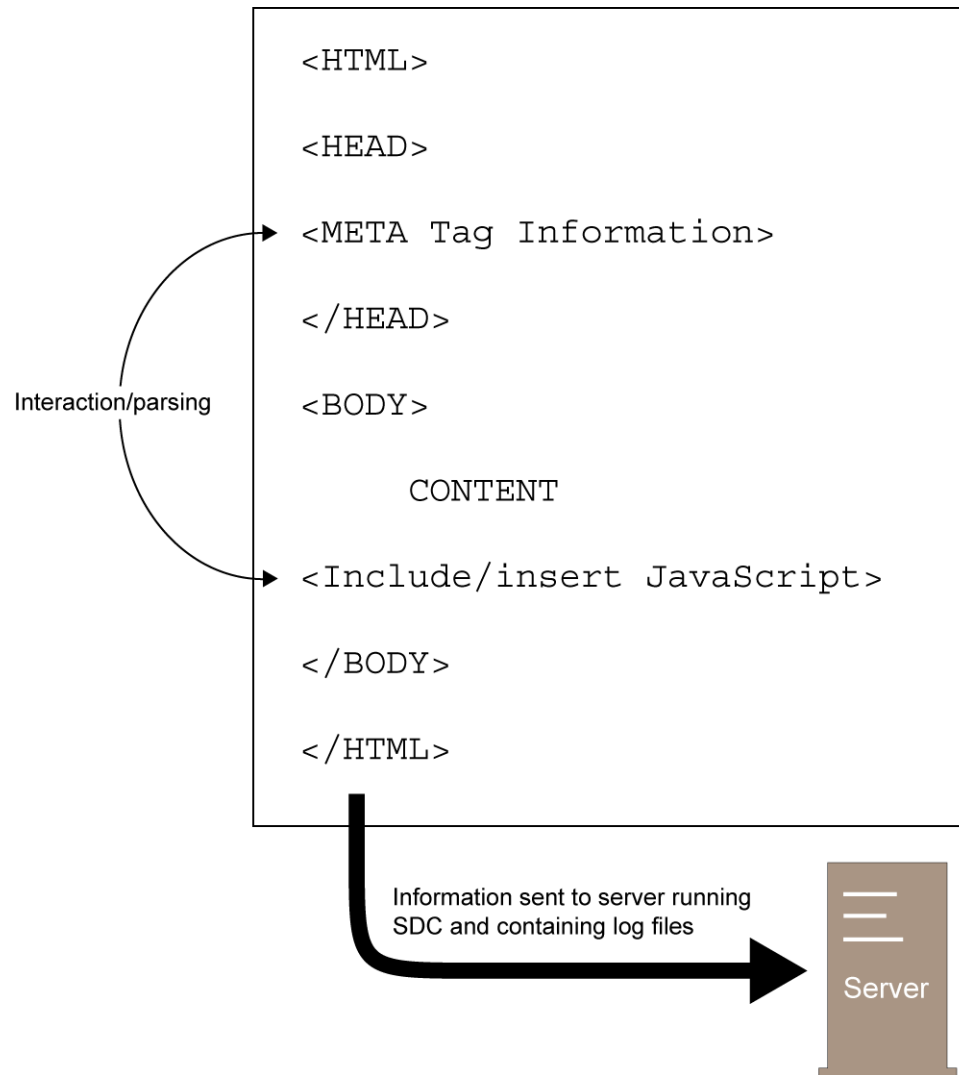


Figure 2-1. Overview of JavaScript interaction with META tags

General META Tag Information

The general syntax of the supported META tag is as follows:

```
<META NAME="name" CONTENT="content">
```

The META tag must be included between the <HEAD> </HEAD> tags.

The **name** represents the parameter name.

The **content** represents the parameter value.

Many META tags allow more than one parameter. Multiple parameters are separated by a semicolon “;” as shown in the following example:

```
<META NAME="name" CONTENT="content1;content2;content3...">
```

META Tag Descriptions

The following subsections discuss the META tags that you will need to add or modify on your web pages.

Note that all of the META tags that you will use begin with WT., for example, WT.cg_n, which you can use to track content groups.

To track content groups

WebTrends can report on visitors according to the content group and content subgroup of the pages they visit. Your site must capture the content group and subgroup in a parameter when the page is visited, and then pass the parameter values to WebTrends. Content subgroups are optional.

The following META tags track content groups.

```
<META NAME="WT.cg_n" CONTENT="Name">
```

Defines the name of the content group.

```
<META NAME="WT.cg_s" CONTENT="subName">
```

Defines the name of the content subgroup.

The WT.cg_s tag is optional.

Example: single content group and subgroup

A university web site reports the number of visitors to its pages. The web site denotes the content group DegreeReq for each page that describes the requirements. The web site then assigns a subgroup designation for each page about a particular field of study, for example, Lit for Literature.

Using the WT.cg_n and WT.cg_s tags, your META tag would look like this:

```
<META NAME="WT.cg_n" CONTENT="DegreeReq">
<META NAME="WT.cg_s" CONTENT="Lit">
```

Example: Multiple content groups

Building on the single content group example, the Admissions Department is also interested in visitors to these pages. An additional group can be added so that the pages are reported for both content groups. Note that Math has been added as a subgroup. Multiple content groups and subgroups are delimited by semi-colons.

```
<META NAME="WT.cg_n" CONTENT="DegreeReq;Admissions">
<META NAME="WT.cg_s" CONTENT="Lit;Math">
```

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this – assuming that this web site sells communications equipment.

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//www.biguni-
versity.edu&WT.cg_n=DegreeReq;Admissions&WT.cg_s=Lit;Math....
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track servers

If your site is hosted on multiple servers, a cluster, or a farm, and you want to evaluate the performance of your load balancer, WebTrends can track page views for each server. To do so, populate the following META tag to all pages on each server:

```
<META NAME="WT.sv" CONTENT="name">
```

Defines the name of the machine that serves up the web page.

If you have two servers (Server1 and Server2), you would make two copies of the META tag and designate `CONTENT="Server1"` for deployment to pages on the first server and `CONTENT="Server2"` for deployment to the same pages on the second server.

For a server farm, you can extract the value of the built-in server name and dynamically assign it to the META tag.

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.sv=Web%20Server . . .
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track marketing campaigns

WebTrends can report visitor activity that relates to a marketing campaign. You can place the following META tags on the landing page to identify the name of the campaign and the type of campaign.

```
<META NAME="WT.mc_n" CONTENT="Name">
```

Defines the name of the marketing campaign.

```
<META NAME="WT.mc_t" CONTENT="type">
```

Defines the type of marketing campaign.

Regarding marketing campaign names, choose name that are short and meaningful. Long names may exceed IE query string limits.

Landing Page

The landing page is the first page that visitors see when they visit your site. Normally, this is your home page, but for effective marketing campaign tracking, you can bring visitors to a page exclusively used for your marketing campaign.

Example marketing campaign

To attract new students, a university launches a marketing campaign by sending recruitment e-mail to all graduating high school seniors in a metropolitan area. The e-mail links to a special landing page in the university's web site, containing the following META tag to track marketing campaigns.

```
<META NAME="WT.mc_n" CONTENT="Recruit">  
<META NAME="WT.mc_t" CONTENT="e-mail">
```

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1  
GET /ads/default.asp redir=products&ad=http%3A//www.biguni-  
versity.edu&WT.mc_n=Recruit&WT.mc_t=e-mail . . . .
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track profile/subprofile generation

If you are using parent/child profiles, the following META tag allows you to determine which child a hit goes to:

```
<META NAME="WT.sp" CONTENT="profile name">
```

Defines the name of the profile or subprofile.

Refer to the WRS help files for more information regarding parent/child profiles.

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1  
GET /ads/default.asp redir=products&ad=http%3A//  
www.phonedealet.com&WT.sp=Wireless%20Phones . . .
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track revenue

To track eCommerce revenue, you must populate the META tags on your confirmation pages. Your web site captures transaction information, typically using an order form. You must pass the values from the form to a META tag so that WebTrends can track the transactions, aggregate them, and include them in your reports.

To track revenue, you must include the following META tags:

```
<META NAME="wt.pn" CONTENT="ProductName">
```

Defines the name of the product. Semicolons can be used to denote more than one name.

```
<META NAME="wt.pc" CONTENT="ProductCategory">
```

Defines the category of the product. Semicolons can be used to denote more than one category.

```
<META NAME="wt.tx_t" CONTENT="type">
```

Defines the type of order. Semicolons can be used to denote more than one type.

```
<META NAME="wt.tx_u" CONTENT="units">
```

Defines the quantity purchased. Semicolons can be used to denote more than one quantity.

```
<META NAME="wt.tx_s" CONTENT="subtotal">
```

Defines the total cost of an order. Semicolons can be used to denote more than one total cost. However, do not pass a dollar sign (\$) or comma(,) in the subtotal variable.

Example of multiple usage

You can pass multiple orders to the variables in the META tags by using a semicolon delimited list. For example, the following tag represents two products:

```
<META NAME="wt.pn" CONTENT="wireless Phones;Standard Phones">
```

Example log file

After the JavaScript interacts with these META tags, it generates log files that may look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedea1er.com&WT.pn=NewCoo1
XY2&WT.pc=ce11phone&WT.tx_t=Credit%20Card&WT.tx_u=2$WT.tx_s=130.00
....
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

Note: In the log file %20 is an ASCII representation for a blank space.

To track shopping cart activity

This section discusses the META tags that track shopping cart activity. At analysis, these tags are interpreted as steps along the path that leads to a successful completion of the shopping activity. There are four steps in all. Each page in the shopping cart process needs a corresponding META tag.

Note: Your web site may not require the use of all the tags listed. For example, in some cases, WT.si_x may not be needed.

Step #1 – product view

Your first web page for shopping cart activity uses the following META tags:

```
<META NAME="WT.si_n" CONTENT="name">
```

Defines the name of the shopping cart activity. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_p" CONTENT="CartView">
```

Defines the name of the particular step in the shopping cart process. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_x" CONTENT="stepPosition">
```

Defines the numeric step position in the shopping cart activity. Semicolons can be used to denote more than one position.

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.si_n=ShoppingCart&WT.si_p=CartView&WT.si_x=
1 .....
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

Step #2 – addition to cart

Your second web page for shopping cart activity uses the following META tags:

```
<META NAME="WT.si_n" CONTENT="name">
```

Defines the name of the shopping cart activity. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_p" CONTENT="CartAdd">
```

Defines the name of the particular step in the shopping cart process. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_x" CONTENT="stepPosition">
```

Defines the numeric step position in the shopping cart activity. Semicolons can be used to denote more than one position.

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.si_n=ShoppingCart&WT.si_p=CartAdd&WT.si_x=2
.....
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

Step #3 – check out

Your third web page for shopping cart activity uses the following META tags:

```
<META NAME="WT.si_n" CONTENT="name">
```

Defines the name of the shopping cart activity. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_p" CONTENT="CartCheckout">
```

Defines the name of the particular step in the shopping cart process. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_x" CONTENT="stepPosition">
```

Defines the numeric step position in the shopping cart activity. Semicolons can be used to denote more than one position.

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1  
GET /ads/default.asp redir=products&ad=http%3A//  
www.phonedealet.com&WT.si_n=ShoppingCart&WT.si_p=CartCheckout&WT.s  
i_x=3 . . . .
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

Step #4 – purchase

Your fourth web page for shopping cart activity uses the following META tags:

```
<META NAME="WT.si_n" CONTENT="name">
```

Defines the name of the shopping cart activity. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_p" CONTENT="CartBuy">
```

Defines the name of the particular step in the shopping cart process. Semicolons can be used to denote more than one name.

```
<META NAME="WT.si_x" CONTENT="stepPosition">
```

Defines the numeric step position in the shopping cart activity. Semicolons can be used to denote more than one position.

Note: For purchase (this step), you also need to include information for tracking revenue

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.si_n=ShoppingCart&WT.si_p=CartCheckout&WT.s
i_x=4 ....
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track Advertising Views

Visitors often view advertisements that they don't necessarily click on. You can use Advertising Views to determine the number of visitors to your web site who view particular ads. With this feature you can produce advertising reports for each of your clients.

If you are selling advertising space on your web site, for example, you can collect traffic statistics to help determine pricing schedules. If you use the Advertising filter in your profile, you can show traffic patterns for a specific advertisement (such as busiest time of the day or busiest day of the week), as well as demographic information (such as the top viewing or clicking organizations or countries).

The following META tag tracks advertising views:

```
<META NAME="WT.ad" CONTENT="name">
```

Defines the name of the advertisement viewed on this page.

Multiple Advertising Views can be designated by using semicolons.

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.ad=Weekend%20Special%20Rate ...
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track Advertising Clicks

When a visitor to your site clicks on an ad, that action is referred to as an *Ad Click*. The following META tag tracks advertising clicks:

<META NAME="WT.ac" CONTENT="name">

Defines the name of the advertisement clicked to reach a particular web page. The Ad Click must contain an external redirect back to the client. The redirect needs to include the necessary code to generate a hit to the SDC server.

Multiple Advertising Clicks can be designated by using semicolons

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1
GET /ads/default.asp redir=products&ad=http%3A//
www.phonedealet.com&WT.ac=Weekend%20Pointer ...
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track customized URLs

WebTrends reports only the base URL when it compiles reports on pages identified by their URLs. "Top Pages" and "Page Views Trend" are reports that use page URLs. These reports could become overwhelming and meaningless because URL parameters are used to carry many different kinds of information in addition to dynamic page identification. There can be many variations to a single page URL. The parameters make it seem as if there are many unique URLs when there is only one, the base URL.

Dynamic pages are an exception. With dynamic pages, counting each URL with different parameters is required. By customizing a URL, you can track dynamic pages by changing the URL before it is passed to WebTrends.

For the following three site types, you will need to customize a URL before sending it to WebTrends:

1. A site using a small number of filenames and serve content based on parameters.
2. A site using session or user ID information in the path or the filename of the site.
3. A site using cryptic URLs that can be translated into more meaningful information.

In each of these cases, you should customize the URLs to get full use of the WebTrends reports. Without customization, the reports could contain too much data, rendering them useless.

A site using a small number of filenames where the content is served based on parameters.

Web sites that use dynamic information to build content introduces a challenge for reporting. WebTrends removes the parameters to prevent too many URLs from overshadowing specific data. If this removal was not done, a user ID string or a session ID string could possibly generate over a million unique URLs, as well as overshadow any useful data in the Top Pages report.

For example:

Site www.asite.com is selling CDs. The Webmaster knows that they will have to add many CDs to the site on a constant basis. Instead of creating a new page for each CD, the Webmaster sets up a template page and uses a database to store specific content for each CD. He also sets up a shopping cart to allow multiple CDs to be purchased. The corresponding URL could look something like:

```
http://www.asite.com/  
browse.asp?UID=14&Cat=Rock&Artist=Alice+In+Chains&Album=Jar+Of+  
Flies
```

Without modifying the URL, the Top Pages section would only show:

```
http://www.asite.com/browse.asp
```

In this circumstance some of the parameters have the data needed to get useful information in the report. Using scripting, usually at the server level when the template is being served, the URL can be modified to look something like:

```
http://www.asite.com/Rock/Alice_In_Chains/Jar_of_Flies.asp
```

The URL would appear similarly in the Top Pages, and you can modify the URL to display the parameters that are most useful to you.

A site using session or user ID information in the path or filename of the site

The www.asite.com Webmaster has decided to engineer a shopping cart application that serves the User ID variable as part of the site's file structure instead of serving it as a parameter. The URL could look like:

```
http://www.asite.com/<UID_String>/Rock/Alice_In_Chains/  
Jar_Of_Flies.asp
```

The UID string would be different for each user visiting the site, and would generate a large number of unique pages in the Top Pages section of the report. The report wouldn't show how many hits to a specific album, but would show a new page for each user that visits the album page.

To resolve this issue, the Webmaster can generate a script to “filter” the UID String from the URL. The URL would now look like:

```
http://www.asite.com/Rock/Alice_In_Chains/Jar_Of_Flies.asp
```

A site using a content management system that products cryptic URLs

The site www.asite.com has not become large enough to make standard content management too much of a problem. The Webmaster has purchased Vignette Story-server. When the server is implemented and running, the URLs changes to something like:

```
http://www.asite.com/Rock/0,45,187,00.HTML
```

The URL gives no indication of the page content. While such a URL doesn't affect the site visitor, it makes for uninformative WebTrends report data.

Translating the URL provides more useful report data. You can use scripting the template to translate the URL numbers into meaningful information from the Vignette Database. Once translated, the URL would look like:

```
http://www.asite.com/Rock/Alice_In_Chains/Jar_Of_Flies.HTML
```

To translate the URL, the Webmaster or Web Developer scripts a URL string and passes it to WebTrends.

Other content management systems can produce URLs similar to Vignette. The best way to determine whether your URLs need translation is to browse through the site and review the URLs for meaningful content.

Syntax for META tags

The following META tags allow you to customize URLs:

```
<META NAME="DCS.dcspro" CONTENT="protocol">
```

Assigns the information to the cs-version field of the log file.

```
<META NAME="DCS.dcsref" CONTENT="referer">
```

Assigns the information to the cd-referer field of the log file.

```
<META NAME="DCS.dcSSIP" CONTENT="domain">
```

Assigns the information to the cs-host field of the log file.

```
<META NAME="DCS.dcsuri" CONTENT="uri-stem">
```

Assigns the information to the cs-uri-stem field of the log file.

```
<META NAME="DCS.dcsqry" CONTENT="uri-query">
```

Assigns the information to the cs-uri-query field of the log file.

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3SVC3 web1 192.168.1.1  
GET /ads/default.asp redir=products&ad=http%3A//  
www.phonedealet.com .... http://www.example.com/exampledir/  
homepage.html ...
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

To track page titles

You may want to modify a page title before sending it to WebTrends in the following cases:

- You are dealing with dynamic content pages identified by URL parameters, and the page title represents the title of the base URL page rather than the dynamic content page.

Unless you modify the page titles, all pages will have the same title in the reports.

- All pages have been assigned the same title, for reasons of style or company policy.

Even though URLs are displayed in addition to page title, the entire URL can't be depended upon to distinguish one page from another.

Use scripts to change the title to something that reflects the content of the pages so that you can identify them in reports. Next, pass the customized page titles to WebTrends, using the following META tag:

```
<META NAME="WT.ti" CONTENT="title">
```

Defines the name of the title for this page.

Example log file

After the JavaScript interacts with this META tag information, it generates log files that look something like this:

```
2001-03-04 00:08:18 proxy7.hotmail.com - w3svc3 web1 192.168.1.1  
GET /ads/default.asp redir=products&ad=http%3A//  
www.phonedealet.com&WT.ti=Advertising ...
```

The italicized text represents the META-related order information captured by the JavaScript and placed in the log file.

Chapter 3

Deployment of the META Tags

Deployment of the code can be accomplished in several different ways: copying appropriate tag versions to each of your Web site pages, using one or more include files with include statements on your pages, or using server-side includes on appropriate Web servers. Deployment depends on your needs, your maintenance practices, and the programming resources available to you.

Once you have deployed the code to your site, you can begin viewing reports of your visitor activity and of the revenue your site generates.

Copy the code to each page

Copy the same code or individually modified copies of the code to your Web site pages. To minimize the impact on your website, it's advisable to place the code as close as possible to the `</BODY>` tag.

Using Include Files

When you want to add the same code to numerous pages, you can use an include that contains the JavaScript code. Using an include has a definite advantage when you must modify or upgrade the code. When you modify the code in the include file, it is automatically updated to all pages that have the include statement. Include the JavaScript code within the `<BODY>` and `</BODY>` tags.

Server-Side Include Files

Server-side includes (SSI) are enabled by default on Internet Information Server and Apache Web servers. You can either configure the server to run SSI on all files with the extensions you use for your Web pages (.htm, .HTML), or you may need to change your page extensions (to .stm, .shtm, and .sHTML, for example).

Set up the include file and the include statements:

1. Place the include file with the JavaScript code in it in a location accessible to every page of your site.

2. Place an include statement on all of your Web site pages. Be sure to use the correct file extensions.

For example, an include file named code_include.inc is located in the directory mysite. Place the following include statement on your Web pages:

```
<!--#include virtual="/mysite/code_include.inc"-->
```

JavaScript Source Includes

In JavaScript 1.1 and later, you can include the contents of an external file on every page of your Web site. This method uses the Script Tag to start the Script but then uses a file for the actual Script. Since the JavaScript is already open you must remove the HTML tags and the open Script tag from the JavaScript Tag. This will be the first 7 lines in the Script as well as the last part of the Script, as per the following instructions:

1. Use the file you save the WebTrends JavaScript. For example, place the file in the mysite directory and name it SDC.js:

```
/mysite/SDC.js
```

2. Remove the SCRIPT tags from the WebTrends JavaScript:

Remove all lines up to and including the last open Script tag:

```
<SCRIPT LANGUAGE="JavaScript">
```

Remove the last five lines from the bottom, including:

```
</SCRIPT>
```

Note: *The JavaScript will be called from within the SCRIPT tags in the call to SDC.js you place on your Web pages in Step 4.*

3. Cut the <NOSCRIPT>... noscript tag contents ...</NOSCRIPT> block from the WebTrends JavaScript. It is the fourth, third, and second lines at the bottom. Place it after the call to SDC.js you place on your Web pages in Step 4.
4. Place a JavaScript call to SDC.js and the <NOSCRIPT> block on every page of your Web site:

```
<SCRIPT src="/mysite/SDC.js" language="javascript"  
TYPE="text/javascript"></SCRIPT>  
<NOSCRIPT>... noscript tag contents ...</NOSCRIPT>
```

Header and Footer Templates

If your setup uses header and footer templates, you can place the JavaScript code in the footer template. Place it as close as possible to the `</BODY>` tag

However, an eCommerce confirmation page, and the confirmation page only, requires the revenue tracking code. If you use a footer template containing the tracking code, the footer on the confirmation page must be different from the footer on the other pages.

Web Page Editors

A number of HTML editors actually modify your HTML code and can break JavaScripts. Make sure that your HTML editor does not modify your HTML or WebTrends JavaScript in any way.

Generating and implementing WebTrends JavaScript using a common template

Customers with multiple reports can avoid implementing and maintaining multiple sets of WebTrends JavaScripts by using a template approach.

Instead of maintaining separate sets of the JavaScript code and adding them to individual pages or Web applications, the JavaScript code administrator could store all variable elements of the JavaScript code in a database table. A function can then be used to insert these values in a common JavaScript code template at page generation time, as well as to make any other required customization to the WebTrends JavaScript.

This approach eliminates the need to add the JavaScript code manually to the pages that require reporting. It allows the administrator of the JavaScript code to maintain as little as a single JavaScript code generating function and a database table containing JavaScript code parameters, and generally provides a great degree of flexibility over addition of the JavaScript code to HTML pages.

