

HitBox[®] | PROFESSIONAL

**HitBox Professional V3
Installation Guide**

Installation Overview.....	2
Page Names and Illegal Characters	2
Changing Default Settings for Page Names	2
Designating Content Groups	3
Changing Default Settings for Content Groups	3
Coding Multi-level Content Groups	3
Basic Installation Instructions.....	4
Installing HitBox Professional Code with the Installation Wizard	4
The Remove Code Option	5
The Automatic Installation Option	5
The Manual Installation Option	6
Installing HitBox Professional Code Manually.....	7
Commerce Tracking Code	8
Installation for Frames	9
Advanced Installation Instructions	10
Installation for Dynamically Generated Pages.....	10
About the Backslash	10
Installation of Commerce Features in Dynamically Generated Pages.....	11
Installing HitBox Professional Code for Sites Built with Macromedia Flash.....	11
Integration of HitBox Professional JavaScript with Flash ActionScript.....	12
ActionScript Code Samples.....	13
Implementation of HitBox Professional Code for Flash Sites	13
Sample Scripts	14
Perl Example	14
Cold Fusion Example.....	15
PHP Example	17
ASP Examples.....	20
Simple Method	20
Advanced Method.....	20
Main Page Code/Sub-Page Code	21
Appendix A: hitboxe.asp	22
Appendix B: Simple ASP Example.....	24
Appendix C: Advanced ASP Example	25

Questions about the material in this installation manual? Suggestions for how to improve the manual in future revisions? Send an e-mail to manual@websidestory.com, and our manual revision team will consider your suggestions.

Note: You will not receive a response to this e-mail.

Installation Overview

Installation of HitBox Professional is as simple as embedding code (provided by WebSideStory) into the Web site pages for which you want HitBox Professional to measure traffic. This code is customized for your site and includes SSL code for secure pages, as well as Flash code for tracking visitor interaction with Flash objects.



NOTE: To continue with this installation, you must have an administrator-level password to be able to logon to HitBox Professional's Site Tools and Edit Account area.

You insert a unique name into the HitBox Professional code on each page, so that HitBox Professional can distinguish one page from another. You may also group sets of pages using a content grouping variable. You can add the unique page and content group name and paste this code into your site pages manually, with custom scripting, or using the Installation Wizard.

Page Names and Illegal Characters

Before you begin coding, make sure you have access to all your Web site pages. You should also determine a unique name for each of those pages. Keep these items in mind when you are defining page names:

- The name cannot be longer than 50 characters.
- You cannot use non-ASCII characters in page names.
- Use + instead of spaces in the name and do not use any of the following characters in the name

; \ ' : | \ / # & % \$ > < ! ^ * ~ ' "

They may interfere with the operation of the HitBox Professional code. Also, you may not use any accented characters or the associated HTML code used to create accented characters (such as é, ñ, á, or é á ñ).

- If a page includes a product you want to track, you may want to include an identifier for the product in the page name (such as Catalog+Rolex515).

The page name variable in the HitBox Professional code is `_pn`. The default assignment for the variable is "PUT+PAGE+NAME+HERE". In the code, it appears this way:

```
var _pn="PUT+PAGE+NAME+HERE";
```

If you do not define a page name within the HitBox Professional code, the Page Name variable will default to the page title within the HTML `<title></title>` tags. If you choose not to name your page, make sure to leave the default page name text "PUT+PAGE+NAME+HERE" in the code.

Changing Default Settings for Page Names

You may set the page name variable to default to the file name of the site page (such as, `aboutus.html`), rather than the page title. To do so, change the variable assignment `_pndef="title";` to `_pndef="index.html";` (or `_pndef="home.html";` or whatever file

name the default document name is for your server) within the HitBox Professional code. When this variable has been changed, HitBox Professional will use the actual file name of the page instead of the page title if you do not define a page name with the page name variable.



NOTE: You must define your default file name when you change this variable so that HitBox Professional can measure visits to URLs such as `http://www.hitboxProfessional.com` correctly.

Designating Content Groups

With HitBox Professional, you can group site pages together that contain similar types of content, such as download pages, product pages, corporate information pages, and so on. Group your pages together using the content group variable in the HitBox Professional code. Replace the content group default text "CONTENT+CATEGORY" with a content group for each page that you wish to be part of each content group.

The content variable in the HitBox Professional code is `_mlc`. The default assignment for the variable is "CONTENT+CATEGORY". In the code, it appears this way:

```
_mlc="CONTENT+CATEGORY";
```

For example, all your pages that provide product information could be identified with the content group "Products" within the HitBox Professional code. Content groups should follow the same parameters as page names, listed above.

Note that you are not required to add content group names to your pages. If you choose not to specify content groups for your pages, you must leave the text "CONTENT+CATEGORY" in the code.

If you do not define a content group for a page, then the content group variable will default to the full directory path of the site page. For example, if the URL for your page is `http://www.websidestory.com/about/company/location.htm`, the full directory path is `/about/company`

Changing Default Settings for Content Groups

The default assignment for the variable `_ctdef` is "full". If you want the content group variable default to display only the current directory in which the page resides, change the assignment `_ctdef="full";` to `_ctdef="current";` in the HitBox Professional code. For example, to show only the `/company` directory as in the above example, use `_ctdef="current";`

Coding Multi-level Content Groups

The content grouping variable `_mlc` in the WebSideStory code allows you to define up to two content levels for your site pages. Replace the text "CONTENT+CATEGORY" in the content variable with a content group name for each page that you wish to be part of each content group. Separate content group levels with a forward slash /

For example, if your site contains a page on autos, and that section contains the subsections store, accessories, and safety, you may define content levels for each of those sections. Pages in the top-level auto section would be given the content group `/Auto`, the pages under the store section would be given the group `/Auto/Store`, the accessories section `/Auto/Accessories`, and so on.

The initial forward slash / is placed at the beginning of each content group (/Auto/Store not Auto/Store). There should be no forward slash at the end (/Auto/Store not Auto/Store/). Content groups should follow the same naming conventions as page names.

Basic Installation Instructions

This section will give you instructions on the basics of installing HitBox Professional code, both with the Installation Wizard as well as installing the code by hand. Please refer to Advanced Installation Instructions for information on how to add HitBox Professional code to frames, dynamically generated pages, and to Web sites coded with Flash Macromedia technology.

Installing HitBox Professional Code with the Installation Wizard

The following provides you with instructions on installing and using the Installation Wizard. For more detailed information, please refer to the Installation Wizard's help files.



NOTE: The Installation Wizard is designed for adding HitBox Professional code into static HTML pages using a Windows-based computer. If you have Active Server Pages on your site, or other dynamically generated pages, see Installing the HitBox Professional Code without the Installation Wizard for instructions on how to insert the HitBox Professional code manually or with custom programming.

To use the Installation Wizard, you must first download it and install it on your computer.

- Logon to HitBox CenterPoint. with your HitBox Professional account, user name/password, and click on the link to 'Installation Wizard' under Downloads.

The Installation Wizard has an Automatic Installation option and a Manual Installation option. A Remove Code option is also available to assist you in removing old HitBox code from your site pages.

- Automatic Installation copies the code into multiple pages at once and generates a unique page name for each page, based on the title listed in the HTML <title> tag (you must supply content group names for your pages, if any). It also provides an automatic backup copy of your files.
- Using the Wizard's Manual Installation option helps you insert page names and content groups into the code. While it copies the code to your computer's clipboard, it requires you to paste the code into the pages by hand. Further, you must provide both the page and content groups.



NOTE: To use the Installation Wizard to install code, you must be connected to the Internet. You do not have to be connected to the Internet to use the Remove Code option.

The Remove Code Option

1. Double-click the HitBox Professional Installation Wizard icon.
2. Select “I want to remove HitBox Professional code from my pages.” Click 'Next'. The remove code window appears.
3. Use the navigation panel at the top of the window to get to the correct folder that contains your site’s HTML pages. Add the pages you want to remove code from to the Selected Pages list by dragging and dropping them into the list, or highlighting each page and clicking on the 'Add to List' button. Hold down your Control key to select more than one page at a time.
4. When you are done selecting the pages, click 'Remove Code'. A message appears to indicate when the removal is complete.

The Automatic Installation Option

1. Double-click the HitBox Professional Installation Wizard icon.
2. Select “I want to add HitBox Professional code to my pages.” Click Next.
3. Enter your account ID and password. The Installation Wizard uses this information to contact the HitBox Professional server and retrieve the customized code for your site.
4. Select 'Automatic Installation'.
5. Select whether you want the code copied into the top or bottom of the file. Click Next'.



NOTE: Pasting the code into the top of the page is recommended. Your statistics will be more accurate and partial loads of the page will be tracked. HitBox Professional code should not cause any page layout changes, as long as your visitors are viewing your Web site using Internet Explorer, version 4.x or higher, or Netscape Navigator, version 3.x or higher.

6. Use the navigation panel at the top of the window to get to the correct folder that contains your site’s HTML pages. Add the pages to the Selected Pages list by dragging and dropping them into the list, or highlighting each page and clicking on the 'Add to List' button. Hold down your Control key to select more than one page at a time. The Wizard will only install HitBox Professional code into pages that have been added to the Selected Pages list.
7. When a page is added to the list, a page name is automatically generated, based on the title of your page within the HTML `<title>` tag. You can change that name by selecting the page within the Selected Pages list and clicking 'Rename Page'.
8. Group your pages by content, if you choose.
9. Type in a name for a backup folder. Click the 'Create Backup Folder' button. The Installation Wizard will create a new, empty folder using the backup folder name you have designated. You are required to create a new folder as the Installation Wizard will overwrite **any and all** existing files (including subdirectories and folders) in any pre-existing folder.
10. If your pages are secure pages, select SSL at the bottom of the window.



NOTE: If you have a set of HTML pages, some that require SSL and some that do not, you can add the SSL HTML pages to the Selected Pages list first, select the SSL option, and insert the SSL version of the HitBox Professional code into those files. Then remove those SSL HTML pages from the Selected Pages list, select the remaining (non-SSL) HTML pages, select the non-SSL option, and insert the non-SSL version of the code into those pages.

11. When you are done selecting the pages, click 'Insert Code'. A message appears to indicate when the installation is complete.

The Manual Installation Option

The Installation Wizard's Manual Installation option does not create backup copies of your files. You may want to create your own backups before beginning this procedure.

1. Double-click the HitBox Professional Installation Wizard icon.
2. Select "I want to add HitBox Professional code to my pages." Click Next.
3. Enter your account ID and password. Installation Wizard uses this information to contact the HitBox Professional server and retrieve the customized code for your site.
4. Select the Manual Installation option and click 'Next'. A window containing your code appears.
5. Enter the name by which you want to identify the first page in the Page Name field. The name that you type here will be inserted into the HitBox Professional code when you select the Copy Code to Clipboard button.



NOTE: When using the Installation Wizard's Manual Installation Option, you do NOT need to use the "+" symbol in place of spaces between words. The Wizard will do that for you.

6. Enter the content group name, if any. This name will also be inserted into the HitBox Professional code when you select the 'Copy Code to Clipboard' button.
7. If the page is a secure page, select SSL at the bottom of the window.
8. Click 'Copy Code to Clipboard'. When you select this button, the Wizard will insert the page name and content group name into the appropriate area of the HitBox Professional code, and will then copy the code to your clipboard.
9. Use your HTML editor to open the specific HTML page that matches the page name, place your cursor where you want the HitBox Professional code to be inserted, and perform the Paste operation to insert the code into your page. It is recommended that you paste the code immediately after the open <body> tag.
10. Return to the Installation Wizard and repeat steps 5-9 for **each separate page** in order for HitBox Professional to track your pages accurately.



NOTE: Be sure to change the name in the Page Name field for each page so that HitBox Professional can track your Web site traffic correctly.

Installing HitBox Professional Code Manually

The following instructions will provide you general guidelines for installing the HitBox Professional code by hand. Installation variations for dynamically generated pages and sites with frames are discussed below.

1. Select 'Edit Account' from the Site Tools and Reporting menu. The Edit Account page contains HitBox Professional universal code for your non-secure pages, universal SSL code for secure pages, error page code for 404-error page templates (or other appropriate templates that measure broken links or files not found), and Flash code for tracking visitor interaction with Flash objects.
2. Select the code you need and copy it to the clipboard. Make sure you have selected **all** the code. An easy way to ensure you've highlighted all the code is to place your cursor inside the code window and use the 'Select All' command (Ctrl+A on your keyboard).
3. Refer to page naming conventions and the list of illegal characters before changing or customizing any of the HitBox Professional code.
4. Open the HTML page in a text editor and paste the code into the file immediately after the open <body> tag.



```
Counter Code
Universal code
<!-- BEGIN WEBSIDESTORY CODE v7.3 -->
<!-- COPYRIGHT 1997-2002 WEBSIDESTORY, INC. ALL RIGHTS RESERVED.
U.S. PATENTED. Privacy notice at: http://websidestory.com/privacy -
-->
<script language="javascript">
var _pn="PUT+PAGE+NAME+HERE"; //page name(s)
var _mlc="CONTENT+CATEGORY"; //multi-level content category
var _cp="null"; //campaign
var _acct="WE5206118ARW07EN3"; //account number(s)
var _pndef="title"; //default page name
var _ctdef="full"; //default content category
var _prc=""; //commerce price
```

Figure 1. The HitBox Professional Code

5. Replace the page name variable ("Put+Page+Name+Here") in the code with a unique name for the page. Use the + symbol in place of spaces. For example, a page with the name "Book Order Form" would be added to the HitBox Professional code as "Book+Order+Form". The code contains the page name variable `_pn` in the code in **two** places. You may want to use your text editor's find and replace command to simplify this step. If you do not define a page name within the HitBox Professional code, the page name variable will default to the page title between the opening and closing <title></title> tags.
6. You may set the page name variable to default to the file name of the site page (such as, `aboutus.html`), rather than the page title. To do so, change the variable assignment `_pndef="title";` to `_pndef="index.html";` (or `_pndef="home.html";` or whatever file name the default document name is for your server) within the HitBox Professional code. When this variable has been changed, HitBox Professional will use the

- actual file name of the page instead of the page title if you do not define a page name with the page name variable.
7. Add the content group, if any, by changing the default assignment ("Content+Category") for the content grouping variable `_mlc`. Follow the same guidelines as the page name variable, replacing any spaces with the + sign. The content grouping variable `_mlc` is also located in *two* places in the code. You might want to use your text editor's find and replace command to simplify this step. If you choose not to define your content grouping variable, then the content grouping variable will default to the full directory path of the site page (for example `/about/company/`).
 8. If you want the content grouping variable default to display *only* the current directory in which the page resides (for example `/content/`) you may change the `_ctdef` variable from the default assignment of "full" thus:

```
_ctdef="full"; (default assignment)
_ctdef="current"; (only current directory)
```

9. To add tracking code for purchases made by visitors to specific pages, make changes to the HitBox Professional code installed on your Web pages. JavaScript variables `_prc` (and optionally `_oid`) should be coded on each page you want to use to trigger the recording of a purchase. The values for these variables are typically populated by dynamically generated pages, such as shopping cart scripts.
10. Note that any variables listed in the code not directly addressed in the installation instructions are either automatically populated by the code itself (such as the account variable), or are currently inactive.
11. Repeat steps 3 through 8 for *every* page on your Web site that you want HitBox Professional to track.



NOTE: If you are upgrading your site pages with a new HitBox Professional account, or new HitBox Professional code, make sure you remove any old HitBox code from your pages. You may use the Installation Wizard's Remove Code feature to easily remove code from your HTML pages.

Commerce Tracking Code

HitBox Professional V3.0 and higher now enables you to track purchases made by visitors to specific pages. To use this feature, changes must be made to the HitBox Professional code installed on your Web pages. JavaScript variables `_prc` (and optionally `_oid`) should be coded on each page you want to use to trigger the recording of a purchase. The values for these variables are typically populated by dynamically generated pages, such as shopping cart scripts.

Your site developer should create a program that resides on your Web server to dynamically insert the HitBox Professional code, replacing the price and order ID variables (`_prc` and optionally `_oid`) with the total price of purchases and a unique ID each time the confirmation page is generated.

In the HitBox Professional code, the variables appear initially with no values assigned. Typically, the code for your page is:

```
var _prc=""; //commerce price
var _oid=""; //commerce order
```

If you want to track purchases on a purchase page after the visitor has selected an item for purchase, assign the `_prc` variable a positive value. For example, for a page selling luxury cars, the value of `_prc` is set to 45000, and the code is: `_prc="45000"`; The value of the purchase generated by a single visit and purchase from that page is then calculated as \$45,000. A total of several prices can be inserted dynamically to populate the `_prc` variable using a value passed by your shopping cart script.

Note that only whole figures--no fractional amounts--are used for the value of `_prc`, and that no currency sign is placed before or after the value. Additionally, the value cannot be a negative number.

The other variable--order ID or `_oid` is optional. When coding your pages, use the `_oid` variable together with the `_prc` variable on the purchase confirmation page, assigning it a unique value. You use `_oid` to indicate that HitBox Professional should not count the same purchase twice, which would artificially inflate reported revenue or purchases.

The `_oid` value is limited to 12 alphanumeric characters, and must not contain any illegal characters.

Installation for Frames

If your Web site uses frames, you should insert the HitBox Professional code only in the content frames, not in the navigation frames, side frames, or frameset. Follow the instructions above to add the code to the pages that will make up the content frames of your site.

Do not place HitBox Professional code in multiple frames. If you put code in more than one frameset, you will dramatically inflate your total view and page view counts, and therefore fail to comply with Internet Architecture Board (IAB) standards. You will also artificially inflate your unique visitors, by 200 percent or more.



NOTE: If you are using an HTML editor—such as FrontPage, Homesite, or Dreamweaver—to add the HitBox Professional code, and you are having difficulties, please visit http://resources.hitbox.com/cgi-bin/page.cgi?techsupport/install_help for assistance.

Advanced Installation Instructions

Below are some more advanced instructions for installing HitBox Professional for dynamically generated pages and installing HitBox Professional code to track visitor interaction with Flash objects.

Installation for Dynamically Generated Pages

In order to install HitBox Professional code within dynamically generated pages, your site developer will need to write a program that resides on your Web server to dynamically insert the HitBox Professional code and replace the page name and content grouping variables in the code with the unique page name and content group at the time that each page is being generated.



NOTE: If you only wish to collect statistics at the template level, insert the HitBox Professional code directly into your template page and replace the page name variable with the template's name and the content grouping variable with the template's content group. All pages using one template will be tracked as one page.

If you want to make changes to the page name and content group default variables (as discussed above), you should make these changes to the HitBox Professional code before you set up your dynamic insertion scripts.

About the Backslash



There is a backslash (\) within the HitBox Professional code that is sometimes deleted by programs that dynamically generate pages. This problem should only occur if you are using a UNIX-based `cgi` script to dynamically build your pages (perl scripts, for example). This is because the backslash is a reserved character in most UNIX shells. If this backslash is deleted, HitBox Professional will **not** count your statistics.

Luckily, there is a very simple fix for this. If you have dynamic pages on your site, and if the program you use to generate your site pages deletes this backslash, insert an **extra** backslash into the HitBox Professional code **before** you insert it in the templates that your program uses to dynamically generate your pages. To see where the backslash resides within the code, see the code sample below (the backslash in question is highlighted).

Adding extra backslashes **only** applies to pages dynamically generated using UNIX-based `cgi` scripts. Before you add extra backslashes you may want to perform a small test on some of your site pages first to see if this extra step is necessary. If the backslash is not deleted, **do not** add an extra backslash to the code. Adding an unnecessary backslash can also create problems. If you do not dynamically generate your pages, and instead paste your HitBox Professional code directly into your HTML pages, you do not need to add an extra backslash.

```
<!-- BEGIN WEBSIDESTORY CODE v7.3 -->
<!-- COPYRIGHT 1997-2002 WEBSIDESTORY, INC. ALL RIGHTS RESERVED. U.S. PATENTED. Privacy
notice at: http://websidestory.com/privacy -->
<script language="javascript">
```

```

var _pn="PUT+PAGE+NAME+HERE"; //page name(s)
var _mlc="CONTENT+CATEGORY"; //multi-level content category
var _cp="null"; //campaign
var _acct="WE5206118ARW07EN3"; //account number(s)
var _pndef="title"; //default page name
var _ctdef="full"; //default content category
var _prc=""; //commerce price
var _oid=""; //commerce order
var _dlf=""; //download filter
var _elf=""; //exit link filter
var _epg=""; //event page identifier
var _mn="wpl79"; //machine name
var _gn="phg.hitbox.com"; //gateway name
var _hcv=68;function _wn(n){return((n.indexOf("NAME")>0&& n.indexOf("PUT")>-1)||
(n.indexOf("CONTENT")>-1&& n.indexOf("CATEGORY")>0));}function _gd(x,w){var _ed=
x.lastIndexOf("/");var _be=(w!="full"?x.lastIndexOf("/",_ed-2):x.indexOf("/");
return(_be==_ed)?"/":x.substring(_be,_ed);}function _gf(x){return x.substring(
x.lastIndexOf("/")+1,x.length);}function _tc(_ml){var _mll=_ml.length-1;var s=
"/";if(_ml.lastIndexOf(s)==(_mll-1)){_ml=_ml.substring(0,_mll-1)}if(_ml.indexOf(
s)!=0){_ml=s+_ml;_mll++;}if(_mll){var a=_ml.substring(1,_mll).indexOf(s);if(a==
-1)return _ml;var b=_ml.substring(a+2,_mll).indexOf(s);if(b==-1)return _ml;if(b
!=-1){if(!b)--;return _ml.substring(0,a+2+b);}return _ml;}return _hl=location;
var _lp=_hl.protocol.indexOf('https')>-1?'https://':'http://';var _zo=(new Date(
)).getTimezoneOffset();function _ps(_ip,_pml){return(_ip&&_wn(_pml))?( (_pndef
=="title"&&document.title!="")?document.title:_gf(_hl.pathname)?_gf(
_hl.pathname):_pndef):_tc(_wn(_pml)?_gd(_hl.pathname,_ctdef):_pml);}function
_pm(m,_fml,h){if(m.indexOf(";")!=-1){_nml=m.substring(0,m.indexOf(";"));_rm=
m.substring(m.indexOf(";")+1,m.length);_fml=_ps(h,_nml)+";"}return _pm(_rm,
_fml,h);}else{return _fml+_ps(h,m);}var _sv=10;var _hn=navigator;var _bn=
_hn.appName;if(_bn.substring(0,9)=="Microsoft"){_bn="MSIE";}var _bv=(
Math.round(parseFloat(_hn.appVersion)*100));if((_bn=="MSIE")&&(parseInt(_bv)==2
))_bv=301;function _ex(v){return escape(v);}var _rf=_ex(document.referrer);
_mlc=_pm(_mlc,"",0);_pn=_pm(_pn,"",1);</script><script language="javascript1.1">
_sv=11;</script><script defer src="http://stats.hitbox.com/js/hbp.js"
language="javascript1.1"></script><script language="javascript">if(_sv<11){if(
document.cookie.indexOf("CP=")>-1){_ce="y";}else{document.cookie="CP=null*";p"+
"ath="; expires=Wed, 1 Jan 2020 00:00:00 GMT";_ce=(document.cookie.indexOf(
"CP=")!=-1)?"y":"n";}if((_rf=="undefined")||(_rf=="")){_rf="bookmark";}x2=
"<img src='+_lp+_gn+_/HG?hc="+_mn+"&hb="+_ex(_acct)+"&n="+_ex(_pn);_x3="&cd="+
"&hv=6' border=0 height=1 width=1">";_ar="&bn="+_ex(_bn)+"&bv="+_bv+"&ce="+_ce+
"&ss=na&sc=na&ja=na&sv="+_sv+"&con="+_ex(_mlc)+"&vcon="+_ex(_mlc)+"&epg="+_epg+
"&hp=u&cy=u&dt=&ln=na&cp="+_ex(_cp)+"&pl=&prc="+_ex(_prc)+"&oid="+_ex(_oid)+" "+
"&rf="+_rf;
document.write(_x2+_ar+_x3);}document.write("<!--");</script>
<noscript></noscript><!--/-->
<!-- END WEBSIDESTORY CODE -->

```

Installation of Commerce Features in Dynamically Generated Pages

The commerce features of HitBox Professional are typically used with an automated script for buying goods or services. To use this feature, changes must be made to the HitBox Professional code installed on your Web pages. JavaScript variables `_prc` (and optionally `_oid`) are coded on each page you want to use to trigger the recording of a purchase. The values for these variables are populated by dynamically generated pages, such as shopping cart scripts.

Installing HitBox Professional Code for Sites Built with Macromedia Flash

HitBox Professional Code for Flash uses JavaScript code tailored for use with Macromedia Flash 4.0+ Web sites and applications. The following instructions will show you how to track your visitors' navigation and interaction with your Flash Web site in the same manner that it is possible to track visitor activity in standard HTML-based Web sites.



NOTE: In order to complete the installation of HitBox Professional code for sites that use Flash objects, you need development knowledge of Flash 4.0, as well as a basic understanding of Flash's ActionScript language.

Integration of HitBox Professional JavaScript with Flash ActionScript

Standard HitBox Professional code is run once as each Web page is loaded, using JavaScript to gather visitor statistics and then sending that data to WebSideStory servers for storage and viewing using the HitBox Professional Viewer. A special version of HitBox Professional JavaScript code is designed for optimized tracking of Macromedia Flash objects. This script is loaded once but run many times. The script encapsulates or wraps the actions of the JavaScript in a function that is then accessed and modified by the Flash ActionScript's `getURL` command. Working with the source `.fla` Flash file in Macromedia Flash development studio, this is accomplished with one line of code in the URL field of the `getURL` command, as follows:

```
javascript:function ("VariableName")
```

In Flash-based Web sites, users navigate to different areas of the site using button-behavior objects. The standard `telltarget` ActionScript command related to these buttons in Flash is part of the `on release` function. In order to use the HitBox Professional Flash tracking capabilities, add the `getURL` command to the `on release` function as well, so that ActionScript will correctly call the HitBox Professional Flash tracking code in the HTML page where the Flash program is embedded.

There are three separate functions that you can call within Flash that will then send data to the HitBox Professional JavaScript:

```
_hbPageView(YourPageName, YourContentCategoryName)
```

```
_hbExitLink(YourExitLinkName)
```

```
_hbDownload(YourDownloadName)
```

For example, if your Web site is about custom software applications, and a particular area of your Flash program discusses application service providers (ASPs), the button leading to the ASP area of your site would contain the line of code relating to the HitBox Professional JavaScript, in addition to the standard `telltarget` and other commands you may use to activate that button on release. The following HitBox Professional ActionScript code in the URL field of the basic `getURL` command would be included for that button:

```
javascript:_hbPageView ("ASP", "/homepage/software+vendors")
```

Here, `_hbPageView` is the name of the HitBox Professional function. This may change depending on the type of function you invoke (page view, download, exit link). `ASP` and `/homepage/software+vendors` are the page name and multi-level content group, respectively, for the area of the Flash site that visitors will be redirected to via the button. These parameters will change for each button.

`_hbExitLink` and `_hbDownload` are parameters only to be used if you wish that button instance to reflect a download or exit from your site. Make sure that you do not combine these functions for one event. For instance, do not try to code a page view and a download for the same event.

NOTE: Make sure that you are tracking visitor interaction with your Flash objects only when there is an event (such as clicking on a button within the Flash object)—**not** when the Flash object loads into the Web page. Tracking the loading of the Flash object itself will result in overcounting of data in some statistics.

ActionScript Code Samples

The following are examples of simple ActionScript code showing differences between an ActionScript without the HitBox Professional Flash tracking implemented, with the Flash tracking implemented, and with Flash tracking implemented to include a download parameter.

Note that in ActionScript the double quotes surrounding ASP and software+vendors must be escaped with backslashes. In addition, the download URL must contain `http://` in the `getURL` command.

Button code **without** HitBox Professional ActionScript Command:

```
on (release) {
    tellTarget ("/vendor") {
        page = "asp";
        play ();
    }
}
```

Button code **with** HitBox Professional ActionScript Command:

```
on (release) {
    getURL ("javascript:_hbPageView (\"ASP\", \"software+vendors\")");
    tellTarget ("/vendor") {
        page = "asp";
        play ();
    }
}
```

HitBox Professional ActionScript **with** a Download Command:

```
on (press) {
    getURL ("javascript:_hbDownload (\"YourDownloadName\")");
}
on (release) {
    getURL ("http://www.yourdomain.com/file.exe");
}
```

Implementation of HitBox Professional Code for Flash Sites

1. Using your HTML editor, open the HTML page where the Flash program will be embedded.
2. Place the HitBox Professional Flash code and define the content group and page name as you normally would with HitBox Professional code for one Web page.
3. Open your Flash object within the Macromedia Flash development studio.
4. Using the URL field, insert the ActionScript `getURL` command for the button instances that will call the special HitBox Professional Flash-tracking JavaScript function so it can dynamically gather information about your visitors. Customize each `getURL` command for each button instance.

5. Compile the .fla file to the .swf format and place the compiled Flash object in your HTML page.

Note that when coding for exit links or download events you must isolate your HitBox Professional ActionScript code by encapsulating it in an `on (press)` event. Make sure the `getURL` commands are in an `on (release)` event of the same button. Failure to do this can yield inaccurate tracking results.



NOTE: To ensure that HitBox Professional tracks your Flash-related statistics accurately, you should always isolate the HitBox Professional `getURL` event from other events. Other functions related to each button action should be placed in a separate ActionScript command from the HitBox Professional function.

Sample Scripts

Here are sample scripts to demonstrate how you can insert the correct HitBox Professional code into your dynamic pages, with the correct page name and content group, as they are being generated.

Perl Example

The following example shows a simple method of incorporating HitBox Professional code into a dynamic environment using perl/cgi scripts. The template is loaded and printed out, with variables in the template swapped for scalars in the perl scripts. The example assumes that you have a template with the HitBox Professional code in it with “markers” where you will be swapping dynamic values.

```
#!/usr/local/bin/perl -w
use strict;
my $page_name = "Airline Tickets"; ## set for substitution later
my $content_category = "Travel" ## set for substitution later
if (param('to_do') eq "results")

    { ShowResults();

    }

sub ShowResults # some subroutine inside your perl script
{
# some database call
my $results = $sth->fetchrow;
## call to template printing subroutine to print new code
&#9;PrintForm($results, "Results");
}

sub PrintForm ## slurps up template, and prints out, substituting
dynamic variables
{
my ($to_show, $page_name) = @_;
open (TEMPLATE, "/templates/form");
my @lines = <TEMPLATE>;
close(TEMPLATE);
print "Content-type: text/html\n\n";
```

```

foreach my $line (@lines)
{
$line =~ s/PUT\+PAGE\+NAME\+HERE/$page_name/g;
$line =~ s/CONTENT\+CATEGORY/$content_category /g;
$line =~ s/_RESULTS_HERE_/$to_show/g;
print ("$line\n");
}
exit;
}

```

Cold Fusion Example

The following example shows a simple method of incorporating HitBox Professional code into a dynamic environment using Cold Fusion. By placing `hitboxe.cfm` somewhere on your server, you can then reference the `<CF_HitBoxE>` tag in your page to automatically insert the appropriate code, page name, and content group name, as shown here:

```

<!-- Passing parameters to template --->
<CF_HitBoxE AccountID="Account Number" Title="Page+Name"
Content="Content+Category">
<P>
This is your text.
</P>

```

The file `hitboxe.cfm` contains the following code:

```

<CFSETTING ENABLECFOUTPUTONLY="True">
<!--
File: CF_HitBoxE
Author: Bryan Kocol (bryan@websidestory.com) Updated by TaG 05/07/2002
Updated by Monty 07/17/2002
Date: 6/30/2000
Description: Auto inserts HitBox Professional code.
Notes: This is based on Raymond Camden's custom tag
"CF_HitBox", but it is used to auto
insert HitBox Professional Version 3 code. You must pass
the Account ID.
---->
<CFIF NOT IsDefined("Attributes.AccountID")
<CFOUTPUT>
<B>CF_HitBoxE Error</B>: You must include the AccountID.
</CFOUTPUT>
<CFABORT>
</CFIF>
<CFPARAM NAME="Attributes.Title" DEFAULT="PUT+PAGE+NAME+HERE">
<CFPARAM NAME="Attributes.Content" DEFAULT="CONTENT+CATEGORY">
<CFOUTPUT>

<!-- BEGIN WEBSIDESTORY CODE v7.3 -->
<!-- COPYRIGHT 1997-2002 WEBSIDESTORY, INC. ALL RIGHTS RESERVED. U.S.
PATENTED. Privacy notice at: http://websidestory.com/privacy -->
<script language="javascript">
var _pn="PUT+PAGE+NAME+HERE"; //page name(s)
var _mlc="CONTENT+CATEGORY"; //multi-level content category
var _cp="null"; //campaign
var _acct="WE5206118ARW07EN3"; //account number(s)
var _pndef="title"; //default page name

```

```

var _ctdef="full"; //default content category
var _prc=""; //commerce price
var _oid=""; //commerce order
var _dlf=""; //download filter
var _elf=""; //exit link filter
var _epg=""; //event page identifier
var _mn="wp179"; //machine name
var _gn="phg.hitbox.com"; //gateway name
var _hcv=68;function
_wn(n){return((n.indexOf("NAME")>0&&n.indexOf("PUT")>-1)||
(n.indexOf("CONTENT")>-1&&n.indexOf("CATEGORY")>0));}function
_gd(x,w){var _ed=
x.lastIndexOf("/");var _be=(w!="full"?x.lastIndexOf("/",_ed-
2):x.indexOf("/"));
return(_be==_ed)?"/":x.substring(_be,_ed);}function _gf(x){return
x.substring(
x.lastIndexOf("/")+1,x.length);}function _tc(_ml){var _mll=_ml.length-
1;var s=
"/";if(_ml.lastIndexOf(s)==(_mll-1)){_ml=_ml.substring(0,_mll-
1)}if(_ml.indexOf
(s)!=0){_ml=s+_ml+_mll++}if(_mll){var
a=_ml.substring(1,_mll).indexOf(s);if(a==
-1)return _ml;var b=_ml.substring(a+2,_mll).indexOf(s);if(b==-1)return
_ml;if(b
!=-1){if(!b)--;return _ml.substring(0,a+2+b);}};return _ml;};var
_hl=location;
var _lp=_hl.protocol.indexOf('https')>-1?'https://':'http://';var
_zo=(new Date
()).getTimezoneOffset();function
_ps(_ip,_pml){return(_ip&&_wn(_pml))?((_pndef
=="title"&&document.title!="")?document.title:_gf(_hl.pathname)?_gf(
_hl.pathname):_pndef):_tc(_wn(_pml)?_gd(_hl.pathname,_ctdef):_pml);}func
tion
_pm(m,_fml,h){if(m.indexOf(";")!=-
1){_nml=m.substring(0,m.indexOf(";"));_rm=
m.substring(m.indexOf(";")+1,m.length);_fml+=_ps(h,_nml)+";";return
_pm(_rm,
_fml,h);}else{return _fml+_ps(h,m);}};var _sv=10;var _hn=navigator;var
_bn=
_hn.appName;if(_bn.substring(0,9)=="Microsoft"){_bn="MSIE";}var _bv=(
Math.round(parseFloat(_hn.appVersion)*100));if((_bn=="MSIE")&&(parseInt(
_bv)==2
))_bv=301;function _ex(v){return escape(v);}var
_rf=_ex(document.referrer);
_mlc=_pm(_mlc,"",0);_pn=_pm(_pn,"",1);</script><script
language="javascript1.1">
_sv=11;</script><script defer src="http://stats.hitbox.com/js/hbp.js"
language="javascript1.1"></script><script
language="javascript">if(_sv<11){if(
document.cookie.indexOf("CP=")>-
1){_ce="y";}else{document.cookie="CP=null*; p"+
"ath=/; expires=Wed, 1 Jan 2020 00:00:00
GMT";_ce=(document.cookie.indexOf(
"CP=")!=-
1)?"y":"n";};if((_rf=="undefined")||(_rf=="")){_rf="bookmark";};_x2=

```

```
"<img
src='"+_lp+_gn+"/HG?hc="+_mn+"&hb="+_ex(_acct)+"&n="+_ex(_pn);_x3="&cd="
+
"1&hv=6' border=0 height=1
width=1">";_ar="&bn="+_ex(_bn)+"&bv="+_bv+"&ce="+_ce+
"&ss=na&sc=na&ja=na&sv="+_sv+"&con="+_ex(_mlc)+"&vcon="+_ex(_mlc)+"&epg=
"+_epg+
"&hp=u&cy=u&dt=&ln=na&cp="+_ex(_cp)+"&pl=&prc="+_ex(_prc)+"&oid="+_ex(_o
id)+" "+
"&rf="+_rf;document.write(_x2+_ar+_x3);}document.write("<\!--
");</script>
<noscript></noscript><!--//-->
<!-- END WEBSIDESTORY CODE -->

</CFOUTPUT>
<CFSETTING ENABLECFOUTPUTONLY="False">
```

PHP Example

The following example shows a simple method of incorporating HitBox Professional code into a dynamic environment using PHP. The example assumes that you have the HitBox Professional code as a template in a local file: /templates/hitbox_code.txt. The example also assumes that your page names are in a database table PRODUCTS having the column NAME.

```
<HTML>
<BODY>
<SCRIPT language=PHP>
/*
Name: dynamic_hitbox.php3
Author: Thaddeus Thacker - WebSideStory, Inc.
(http://www.websidestory.com)
Date: June 26, 2000
Updated: July 17, 2002 (Monty)
Requires: PHP3.0 (and MySQL for the database sample)
*/

/* -----
Declare the ITEM ID variable for this example.
Normally, this would be passed in the QUERY_STRING.
----- */

$item_id = 1 ;

/* -----
Declare the PAGE name and CONTENT TYPE variables.
----- */

$page_name = "" ;
$content_cat = "" ;

/* -----
Database connection variables.
----- */
```

```

$hostname = "dbmachine.yourdomain.com";
$username = "johnshaft";
$password = "superlove";
$dbName = "acme_products";

/* -----
Make connection to database.
----- */

MYSQL_CONNECT($hostname, $username, $password) OR DIE("Unable to connect
to database");
@mysql_select_db("$dbName") or die( "Unable to select database");

/* -----
Select PAGE name variables from table.
----- */

$query = "SELECT name, content_category FROM products WHERE product_id =
$item_id";
$result = MYSQL_QUERY($query);

/* -----
Hopefully the query retrieved 1 record.
----- */

$number = mysql_numrows($result);
IF ($number > 0) :

/* -----
Retrieve Page Name and Content Category from database result set.
----- */

$page_name = mysql_result($result,0,name);
$content_cat = mysql_result($result,0,content_category);&#9;

/* -----
Read in HitBox counter code Template from File.
----- */

$hitbox_string = join('', file('templates/hitbox_code.txt'));

/* -----
Search and Replace variables in HitBox code.
Replace "PUT+PAGE+NAME+HERE" with the $page_name
Replace "CONTENT+CATEGORY" with the $content_cat
----- */

$name_string = "PUT+PAGE+NAME+HERE" ;
$cat_string = "CONTENT+CATEGORY" ;
$hitbox_code = str_replace($replace_string, $page_name, $hitbox_string);
$hitbox_code = str_replace($replace_string, $content_cat,
$hitbox_string);
ENDIF ;

/* -----

```



Print the correct HitBox code into the HTML.

```
----- */  
  
PRINT " ";  
PRINT " ";  
PRINT $hitbox_code;  
  
/* -----  
Close the database connection.  
----- */  
  
MYSQL_CLOSE();  
</SCRIPT>  
</BODY>  
</HTML>
```

ASP Examples

The following examples show two methods of incorporating HitBox Professional code into a dynamic environment using ASP.

Simple Method

By placing the file `hitboxe.asp` (in [Appendix A](#)) in a directory on your server, you can then reference the file in any page or template to automatically insert the appropriate code and page name as shown here:

```
<!-- Passing parameters to template -->
<%
    PageName = "Test Page"
    ContentCategory = "Test Content Category"
    AccountID = "YOUR ACCOUNT NUMBER"
    MachineName = "YOUR MACHINE NAME" %>
<!--#include file = "hitboxe.asp"-->
```

Please refer to [Appendix B: Simple ASP Example](#) to see an example of this simple method of incorporating HitBox Professional code into an `.asp` page.



NOTE: If you do not specify a page name or content category when you insert the code, then the standard HitBox Professional defaults will be used. Your customized code will already include your Account Number (`_acct`) and Machine number (`_mn`)

Advanced Method

You can also use any ASP variables already available in your site page to populate the page name and content variables within the HitBox Professional code. For example, if you wanted to pass the page name and parameters in the URL of the document such as `http://www.yoursite.com/yourpage.asp?pagename=tpage&content=testcontent` you could do so as demonstrated below:

```
<!-- Passing parameters to template -->
<%
    PageName = Request.QueryString("pagename")
    ContentCategory = Request.QueryString("content")
    AccountID = "YOUR ACCOUNT NUMBER"
    MachineName = "YOUR MACHINE NAME" %>
<!--#include file = "hitboxe.asp"-->
```

Please refer to [Appendix C: Advanced ASP Example](#) to see an example of this method of incorporating HitBox Professional code into an `.asp` page.

Main Page Code/Sub-Page Code

Earlier HitBox services used two different sets of code: Main Page and Sub-page code. The Main Page code would be installed on whatever page of your site that you defined as the main page—normally the home or index page. Sub-page code was installed on the rest of your site pages.

HitBox Professional no longer uses Main Page code and Sub-page code. Only one set of code applies to all site pages, called **Universal Code**. HitBox Professional still supports Main and Sub-page code already installed onto sites, so you do **not** need to reinstall code onto all your site pages if your pages are coded with old, HitBox Pro code. But taking advantage of many of the statistics and features that HitBox Professional offers, such as content grouping and cookie detection, **does** require installing the latest version of the HitBox Professional code.

Appendix A: hitboxe.asp

```
<%
'File: hitboxe.asp
'Author: Bryan Kocol (bryan@websidestory.com)
'Date: 11/2/2000 Modified to V7.2 05/08/2002 by TaG
'Description: This file is used to auto insert HitBox Professional code
into ASP pages.
'Notes: You must define the AccountID and MachineName variables in the
pages that reference this file.
IF ISEMPTY(PageName) THEN
PageName = "PUT+PAGE+NAME+HERE"
END IF
IF ISEMPTY(ContentCategory) THEN
ContentCategory = "CONTENT+CATEGORY"
END IF
IF ISEMPTY(AccountID) Or ISEMPTY(MachineName) THEN
Err.Raise 500, "HitBox Professional", "You must define the AccountID and
MachineName variables."
END IF
%>
```

```
        <!-- BEGIN WEBSIDESTORY CODE v7.3 -->
<!-- COPYRIGHT 1997-2002 WEBSIDESTORY, INC. ALL RIGHTS RESERVED. U.S.
PATENTED. Privacy notice at: http://websidestory.com/privacy -->
<script language="javascript">
var _pn="PUT+PAGE+NAME+HERE"; //page name(s)
var _mlc="CONTENT+CATEGORY"; //multi-level content category
var _cp="null"; //campaign
var _acct="WE5206118ARW07EN3"; //account number(s)
var _pndef="title"; //default page name
var _ctdef="full"; //default content category
var _prc=""; //commerce price
var _oid=""; //commerce order
var _dlf=""; //download filter
var _elf=""; //exit link filter
var _epg=""; //event page identifier
var _mn="wp179"; //machine name
var _gn="phg.hitbox.com"; //gateway name
var _hcv=68;function
_wn(n){return((n.indexOf("NAME")>0&&n.indexOf("PUT")>-1)||
(n.indexOf("CONTENT")>-1&&n.indexOf("CATEGORY")>0));};function
_gd(x,w){var _ed=
x.lastIndexOf("/");var _be=(w!="full")?x.lastIndexOf("/",_ed-
2):x.indexOf("/");
return(_be==_ed)?"/":x.substring(_be,_ed);};function _gf(x){return
x.substring(
x.lastIndexOf("/")+1,x.length);};function _tc(_ml){var _mll=_ml.length-
1;var s=
"/";if(_ml.lastIndexOf(s)==(_mll-1)){_ml=_ml.substring(0,_mll-
1)}if(_ml.indexOf
(s)!=0){_ml=s+_ml;_mll++}if(_mll){var
a=_ml.substring(1,_mll).indexOf(s);if(a==
```

```

-1)return _ml;var b=_ml.substring(a+2,_ml1).indexOf(s);if(b==-1)return
_ml;if(b
!=-1){if(!b)b--;return _ml.substring(0,a+2+b);}};return _ml;};var
_hl=location;
var _lp=_hl.protocol.indexOf('https')>-1?'https://':'http://';var
_zo=(new Date
()).getTimezoneOffset();function
_ps(_ip,_pml){return(_ip&&_wn(_pml))?((_pndef
=="title"&&document.title!="")?document.title:_gf(_hl.pathname)?_gf(
_hl.pathname):_pndef):_tc(_wn(_pml)?_gd(_hl.pathname,_ctdef):_pml)};func
tion
_pm(m,_fml,h){if(m.indexOf(";")!=-
1){_nml=m.substring(0,m.indexOf(";"));_rm=
m.substring(m.indexOf(";")+1,m.length);_fml+=_ps(h,_nml)+";";return
_pm(_rm,
_fml,h);}else{return _fml+_ps(h,m);}};var _sv=10;var _hn=navigator;var
_bn=
_hn.appName;if(_bn.substring(0,9)=="Microsoft"){_bn="MSIE";};var _bv=(
Math.round(parseFloat(_hn.appVersion)*100));if((_bn=="MSIE")&&(parseInt(
_bv)==2
))_bv=301;function _ex(v){return escape(v)};var
_rf=_ex(document.referrer);
_mlc=_pm(_mlc,"",0);_pn=_pm(_pn,"",1);</script><script
language="javascript1.1">
_sv=11;</script><script defer src="http://stats.hitbox.com/js/hbp.js"
language="javascript1.1"></script><script
language="javascript">if(_sv<11){if(
document.cookie.indexOf("CP=")>-
1){_ce="y";}else{document.cookie="CP=null*; p"+
"ath=/; expires=Wed, 1 Jan 2020 00:00:00
GMT";_ce=(document.cookie.indexOf(
"CP=")!=-
1)?"y":"n";};if((_rf=="undefined")||(_rf=="")){_rf="bookmark";};_x2=
"<img
src='"+_lp+_gn+"/HG?hc="+_mn+"&hb="+_ex(_acct)+"&n="+_ex(_pn);_x3="&cd="
+
"1&hv=6' border=0 height=1
width=1">;_ar="&bn="+_ex(_bn)+"&bv="+_bv+"&ce="+_ce+
"&ss=na&sc=na&ja=na&sv="+_sv+"&con="+_ex(_mlc)+"&vcon="+_ex(_mlc)+"&epg=
"+_epg+
"&hp=u&cy=u&dt=&ln=na&cp="+_ex(_cp)+"&pl=&prc="+_ex(_prc)+"&oid="+_ex(_o
id)+" "+
"&rf="+_rf;document.write(_x2+_ar+_x3);}document.write("<!--
");</script>
<noscript></noscript><!--/-->
<!-- END WEBSIDESTORY CODE -->

```

Appendix B: Simple ASP Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Simple ASP HitBox Professional Example</title>
</head>
<body>
This page is an example that demonstrates a simple method of
incorporating HitBox Professional code into a dynamic environment using
Active Server Pages.

<P>
This is the HTML or text of your page.
</P>

<!-- Use the code below to automatically insert HitBox Professional-->
<% PageName = "Test Page"
ContentCategory = "Test Content Category"
AccountID = "YOUR ACCOUNT NUMBER"
MachineName = "YOUR MACHINE NAME" %>
<!--#include file = "hitboxe.asp"-->
</body>
</html>
```

Appendix C: Advanced ASP Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Advanced ASP HitBox Professional Example</title>
</head>

<body>
This page is an example that demonstrates a more advanced method of
incorporating HitBox Professional code into a dynamic environment using
Active Server Pages. Any variables that are available to your ASP page
can be used to define the page name and content category. In this
example, the page name and content category are passed in the URL of the
document, such as:
"http://www.yoursite.com/ASP_Example_Advanced.asp?pagename=testpage&cont
ent=testcontent"<br>
If the page name and content categories are not defined the standard
HitBox defaults are used.
<P>
This is the HTML or text of your page.
</P>

<!-- Use the code below to automatically insert HitBox Professional-->

<%
  PageName = Request.QueryString("pagename")
  ContentCategory = Request.QueryString("content")
  AccountID = "YOUR ACCOUNT NUMBER"
  MachineName = "YOUR MACHINE NAME" %>
<!--#include file = "hitboxe.asp"-->

</body>
</html>
```