Unica NetInsight OnDemand

Version 8.1.5

Page Tagging Guide

Copyright

Unica Corporation
Reservoir Place North
170 Tracer Lane
Waltham, MA 02451-1379

Examples and Data

All software and related documentation is subject to restrictions on use and disclosure as set forth in the Unica Software License and Services Agreement, with restricted rights for U.S. government users and applicable export regulations.

Companies, names, and data used in examples herein are fictitious unless otherwise noted.

Trademarks and Patents

Unica, the Unica logo, NetInsight, Affinium and MarketingCentral are registered trademarks of Unica Corporation with the U.S. Patent and Trademark Office. MARKETING SUCCESS STARTS WITH U is a trademark. All other trademarks are the property of their respective owners.

Portions of the software described in this document are covered by U.S. Patent Numbers: 6,317,752, 6,269,325, 6,542,894, and 6,782,390.

The NetTracker and Unica NetInsight products are licensed under the following patents and patent publications: US5,675,510, US6,115,680, US6,108,637, US5,796,952, US6,138,155, US6,653,696, US6,763,386, AU0701813, BR9609217, CA2223919, EP0843946, JP03317705, MX193614, NO09705728, AU735285, CA2246746, CN1174316, CN1547123, CN1547124, DK870234, DE69720186, ES2195170, AU727170, BR9808033, CA2284530, CN1251669, IL131871, JP2000514942, KR341110, NZ337756, WO9641495, EP0870234, EP1130526, EP1168196, US20040078292, WO9810349, US20050114511, US20040221033, WO9843380.

Markup functionality in Unica Marketing Operations is enabled through use of third-party software components from AdLib™ eDocument Solutions and Adobe® Acrobat®. "Powered by AdLib™."

NOTICE: This document contains confidential and proprietary information of Unica Corporation ("Unica"). Use, duplication, or disclosure without the express written consent of Unica Corporation is prohibited.

Table of Contents

Preface

Contacting Unica technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with Unica technical support. Use the information in this section to ensure that your problem is resolved efficiently and successfully.

If you are not a designated support contact at your company, contact your Unica administrator for information.

Information you should gather

Before you contact Unica technical support, you should gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages you see when the issue occurs.
- Detailed steps to reproduce the issue.

Contact information for Unica technical support

For ways to contact Unica technical support, see the Unica Product Technical Support web site: (http://www.unica.com/about/product-technical-support.htm).

2 Introduction to Page Tagging for Web Analytics

Introduction

This document describes how to use Unica page tagging to collect information about your Web site for analysis in Unica NetInsight OnDemand. Unica recommends that you read the entire document before deploying page tags.

For information on reporting on information collected by page tags, see the *Unica NetInsight OnDemand Reports User’s Guide*.

What is page tagging?

Page tagging is the method by which Unica NetInsight OnDemand receives raw data.

Basic page tagging involves sending information to Unica NetInsight OnDemand every time a page loads in a visitor's web browser. You can customize page tagging to also send information when a visitor interacts with a tagged page (for example, uses a particular navigation item or starts a video).

You can use direct tags to send data to Unica NetInsight OnDemand when JavaScript rendering is not possible (such as in cell phone applications or Flash widgets). Direct tags are also known as web beacons.

How page tagging works

Page tagging works as follows.

1. In each page on your site that you want to track, you place a *page tag*. A page tag is an HTML reference to the page tag script. If you want to track every page on your site, you can paste the page tag in a common include file such as a footer.
2. The *page tag script* (**ntpagetag.js**) is a JavaScript file that includes functions that collect information about visitors to your tagged pages. It converts the information into name/value pairs that can be read by Unica NetInsight OnDemand. Unica provides you with the page tag script. You customize it as desired and place it on a publicly accessible Web server (typically the same one that serves your site).
3. The page tag script includes a request to the Unica NetInsight OnDemand servers for a one-pixel-by-one-pixel transparent image—the *page tag image*. When a request for the image file is sent to the Unica NetInsight OnDemand servers, the visitor information that was processed by the page tag script is sent with the request. This is how visitor information is transferred to Unica NetInsight OnDemand. A request for the page tag image is made each time a visitor loads a tagged page or executes a tagged event.

Basic page tagging data, dimensions, and metrics

By default, page tagging collects the following visitor information:

- IP address
- GMT date/time
- Page request with query string
- Referring URL
- Screen resolution
- Screen color depth
- Browser language
- Browser Java support
- User agent
- Cookie information

This information appears in your reports through the default dimensions and metrics. These are the default dimensions:

- Page
- Entry Page
- Exit Page
- Referrer
- Browser
- Platform
- Geographic Data (Country, City, Time Zone, Organization, etc.)
- Date
- Time
- Day of the Week

These are the default metrics:

- Number of visitors
- Number of views
- Number of visits
- Number of visitors
- Number of repeat visitors
- Total time online
- Average viewing time
- Average visit duration
- Views per visit

You can customize page tags to collect data for other dimensions.

What to do when JavaScript is disabled

Because page tagging relies on JavaScript, JavaScript must be enabled in your visitors' browsers in order for you to track them. For visitors with JavaScript disabled, you can include an HTML noscript tag that requests the page tag image in your tagged pages. You can pass other information with that request, as desired. For example:

```
<noscript>
<img src="http://ondemand.unica
.com/images/ntpagetag.gif?js=0" height="1" width="1" border="0"
hspace="0" vspace="0" alt="">
</noscript>
```

The page tag image URL in this script is an example. Since the URL for the page tag image is different for each Unica NetInsight OnDemand customer, you need to use the URL that your Technical Account Manager gave you.

However, because robots and spiders cannot execute JavaScript, you may find that limiting tracking to visitors with JavaScript enabled is an effective way to separate human from non-human visitors.

Planning your tags

In order to get the most from page tags, you must understand your organization's reporting needs. Unica strongly recommends that you review the reporting needs with Unica Professional Services or Technical Account Management. If Professional Services is not involved, use iterative rounds of testing in a data validation profile to ensure the tags meet the reporting requirements. Technical Account Management can help with the testing process.

3 Basic Page Tagging for Web Analytics

About deploying basic page tagging

Basic page tagging means implementing the Unica NetInsight OnDemand page tag to send information when a page loads in a visitor's web browser. You need the following to get started:

- A copy of the page tag script preconfigured to point to the page tag image on the Unica NetInsight OnDemand servers.
- A sample, basic page tag to include in your pages.
- A sample HTML file that includes examples of more advanced page tag techniques such as event tagging.

Your technical account manager provides you with your copy of the page tag script and any instructions specific to your environment. You can access the other files from Unica NetInsight OnDemand by clicking **Help**>**Product Documentation**>**PageTagging.zip**.

After you have the files, do the following:

1. If necessary, modify the page tag script and then deploy it to your Web server. Where you deploy the script file may vary depending on whether you serve secure content.
2. Tag your web site pages and move the tagged pages to your production environment.

To edit and deploy the page tag script

1. Open the page tag script in a text editor.
2. If you downloaded the page tag script from the documentation site, edit the path to the page tags image file (NTPT_IMGSRC and NTPT_HTTPSIMGSRC) to match the path you received from Unica during account startup.

   If you received the page tag script from your Technical Account Manager, the path will already have been set to your location on the Unica NetInsight OnDemand servers.
3. As desired, make any changes to the required or optional variables in the page tagging script to control which fields are included in the page tag and the default page tag settings.

   Unica does not recommend disabling the default fields (defined in the NTPT_FIELDS variable).
4. Place the page tag script on a publicly accessible Web server. Confirm that the script is publicly accessible.

You are now ready to tag your pages.

To tag pages

1. Examine the sample page tag provided to you by your Unica OnDemand technical account manager. If necessary, modify the path to where you installed the page tags script.

   ```
   <!-- BEGIN: Unica Page Tag -->
   <script language="JavaScript"
   src="/scripts/ntpagetag.js"></script>
   <!-- END: Unica Page Tag -->
   ```
2. Paste the page tag into each page on your site you want to track with Unica NetInsight OnDemand.

   If you want to track the entire site, paste the page tag into the master template or a global include file, such as a footer.
3. Copy your tagged HTML pages or the updated master template or include file to the Web server.

If you do not need to customize page tags on a page-by-page basis or use page tags to track events or retail activity on your site, you are finished configuring your Web site to use page tags. See the *Unica NetInsight OnDemand Reports User's Guide* for details on viewing the data you are now collecting in Unica NetInsight OnDemand reports.

Using page tagging with secure pages

The security of the page tag request is determined by the protocol used to call the page tag image and the page tag script. The page tag script contains calls for the page tag image using both HTTP and HTTPS. The script will automatically use the correct variable to match the protocol of the page from which the request is sent. If you specify a relative path to the page tag script in your page tags, your visitors' browsers will automatically use the correct protocol.

If you will be tagging pages that are all secure or all not secure, use the appropriate protocol for the page tag script. If you will be tagging a mix of pages that are secure and pages that are not secure place the page tag script on the same server as your Web site content and in your page tags use a relative path to the script location.

4 Adding Dimensions to Tags

About adding dimensions to tags

A dimension is an item of content you want to measure. Dimension values determine the rows in a Unica NetInsight OnDemand report. For example, Entry Page and Referrer are dimensions.

There are three categories of dimensions:

- Default dimensions for which basic page tags automatically collect data
- Standard dimensions that Unica NetInsight OnDemand recognizes automatically when you add them to the tag
- Custom dimensions that Unica NetInsight OnDemand must be configured to recognize when you add them to the tag

Direct tags do not have default dimensions.

Default dimensions

For basic page tags and event tags, the default dimensions are determined by the page tag script (ntpagetag.js). You can edit the script to disable some of the default dimensions, although Unica does not recommend it.

For details, see documentation on the NTPT_FIELDS variable.

Direct tags do not have default dimensions because they do not call the page tag script.

Standard dimensions Unica NetInsight OnDemand recognizes automatically

Unica NetInsight OnDemand automatically recognizes information passed via the following standard dimensions. Unica NetInsight OnDemand reports on their values without the need to first configure parameters for them.

Name	Description	Value / Example
ets	Unique floating pointer used by your visitors' browsers to help avoid cached page tag requests. The page tag script sets the value for this dimension automatically.	Number
ev	Type of event	Any text value
lk	Code that tells Unica NetInsight OnDemand to count the page tag request as a link to an external site.	1 lk=1
pv	Code that determines whether or not Unica NetInsight OnDemand counts the page tag request as a page view. A value of 0 does not count the request as a view. A value of 1 counts the request as a view. Use pv=0 when using a page tag to send visit-level data. Since counting the page tag image request as a page view is the default behavior, you usually do not need to use pv=1.	0, 1
rta	Products added to the shopping cart on the current page	List of product SKUs, quantities, and values separated by semi-colons
rtc	Products purchased	List of product SKUs, quantities, and values separated by semi-colons
rti	Order ID associated with the current page	String
rtr	Products removed from the shopping cart on the current page	List of product SKUs, quantities, and values separated by semi-colons

Name	Description	Value / Example
rtt	Revenue associated with the current page	Number
rtv	Products viewed on the current page	List of product SKUs separated by semicolons
sc	Status of the request (HTTP status code). Unica NetInsight OnDemand uses this dimension to determine which requests are errors.	Number sc=404
site	Name of the site (used to filter data into profiles)	String
ts	Unique floating point identifier used by your visitors' browsers to help avoid cached page tag requests. The page tag script sets the value for this dimension automatically.	Number
un	Populates values for Unica NetInsight OnDemand's user dimension.	String
vc	Visit cost (the amount of money spent to drive a visitor to your site for this visit). The value must be a float value and cannot include a currency sign or commas.	Number

Additional standard dimensions for use in direct tags

The dimensions in this table are default dimensions in basic page tags and event tags, and Unica NetInsight OnDemand recognizes them automatically when you use them in direct tags. Some of these dimensions are commonly used in direct tags and others are rarely used.

Name	Description	Value / Example
cd	Color depth of the visitor's web browser	Number cd=32
ck	Cookies and values. The list of cookie name-value pairs must be delimited using semicolons and the entire list must be URL encoded.	String ck=UnicaNIODID%3Dti05Rws-KyD6V50cKhL%BSessionID%3D9816106-8172

Name	Description	Value / Example
lc	Requested page (including query string). The value must be in the form of a URL and must be URL encoded.	Any URL lc=http%3A%2F%2Fblog.company.com%2F-page.html
ln	Language of the visitor's web browser	String ln=en
rf	Referrer to the page	A URL-encoded URL rf=http%3A%2F%2Fwww.google.com%2F-search
rs	Screen resolutions of the visitor's web browser, expressed in width times height.	*NumberxNumber* rs=1280x800
tz	Time zone of the visitor's web browser	URL-encoded time zone value tz=GMT%20-04%3AOO

Example: site dimension

In Unica NetInsight OnDemand, profiles are typically configured to filter data based on the `site` parameter. You give each site you want to analyze a unique `site` parameter value and then Technical Account Management creates one profile per site.

Filtering based on the `site` parameter is preferred to filtering based on the URL (page) for the following key reasons:

- Unlike URLs, the `site` dimension values are unaffected by translation services or caching servers. (The URL of a page that is viewed through a translation service or caching server may no longer contain the originating domain name, and thus will not be filtered appropriately.)
- Using the `site` dimension ensures the filtering will work even if the URLs change over time.

If you want to analyze multiple sites in a combined profile, use a common prefix in the `site` dimension values for those sites (for example, the name of your organization for production servers or *test* for test servers). This enables you to write a simple filter for the combined profile.

About using custom dimensions

You can configure your tags to collect data about custom dimensions and send the data to Unica NetInsight OnDemand. You insert custom dimension data into the page tag using name-value pairs (also called parameters). Your Technical Account Manager then needs to configure the appropriate profiles to recognize the parameter.

For example, if you want to analyze which content authors create the most popular content, you can create an `author` parameter and use it in the page tag query string for all page views. Once your TAM has configured your profiles, the profiles contain a report that shows which authors have the most popular content. You can also use the `author` custom dimension to filter other reports.

Constructing the dimension name-value pair

The custom dimension parameter name must meet the following requirements:

- It must use alphanumeric characters only. (Your technical account manager can define a more user-friendly name to display in report and column names.)
- It must be unique. (Be sure you are not using the name of a default or standard dimension, even if that dimension is not currently used in your tags.)

The custom dimension parameter value must be URL encoded using the encodeURIComponent() JavaScript function.

You can either define the appropriate value for the dimension each time you add the dimension's parameter to a tag, or you can set the dimension parameter to the value of an existing variable on your page. (Setting the dimension to a variable may not be possible for all sites or applications. The exact method to use depends on the scripting language used by the site or application.)

Passing multiple values

Generally, the value in a name-value pair is a single value (for example, a single video title). However, you may want to pass multiple values instead (for example, a list of all ads displayed during the page view). You can use a list of separated values.

To include multiple values in the name-value pair:

- Use a separator to indicate where one value ends and another begins. A comma is the default separator, but you can also use a pipe, colon, or semi-colon.
- Tell your Technical Account Manager that you are passing multiple values for this dimension so he or she can configure your profiles appropriately. The TAM also needs to know the separator you are using.

Example:

```
<script language="Javascript">
var NTPT_PGEXTRA='ads=abchotel,discounttheater,acmecar';
</script>
```

Planning your custom dimensions

Every custom dimension must have a corresponding parameter defined in Unica NetInsight OnDemand. You should carefully consider your reporting needs and plan out the dimensions you need before you start creating them.

There is a limit to the number of parameters your profiles can contain. (The exact limit is defined during the account startup process.) Because of this limit, whenever possible you should create parameters that you can use in multiple scenarios. You can use filters to focus your reports on particular values, as in the following example.

Example: Video and audio content

Suppose your site contains video and audio content and you use event tags to track when visitors start a video or audio file. Because you want to analyze video and audio content separately, you create two event types (`ev=videostart` and `ev=audiostart`). You also want to analyze the title of the file that was started, and you can do that with a custom dimension. Your first thought might be to create one dimension for videos and one for audio files, but you will have more reporting options and keep the number of parameters lower if you use one dimension for both video and audio titles (for example, `title`).

If you create a `title` dimension, you can create the following reports:

- Report showing all pages on which visitors started a video and which videos they started (Page Summary with `title` dimension added and a filter of `ev=videostart`)
- Report showing all pages on which visitors started an audio file and which audio files they started (Page Summary with `title` dimension added and a filter of `ev=audiostart`)
- Report showing the most popular video and audio content (Parameter Summary for `title`; if you use `title` for other event types also, such as banner ads, add filters `ev=videostart` and `ev=audiostart`)
- Report showing the most popular video content (Parameter Summary for `title` with filter `ev=videostart`)
- Report showing the most popular audio content (Parameter Summary for `title` with filter `ev=audiostart`)

Adding a dimension to a tag

Before adding a custom dimension, see if the data is already part of the page tag. For example, the referrer value or the page URL may already contain the information. In that case, you do not need to modify the tag. Instead, pass the information along to your Technical Account Manager so your profiles can be configured to analyze the data.

In most cases, you need to insert the dimension data into the tag using the dimension's parameter. How you add the parameter depends on the type of tag.

Basic page tags—global dimensions

If the dimension is one that applies to every page, add the dimension's parameter to the NTPT_GLBLEXTRA variable in the page tag script (ntpagetag.js). Separate name-value pairs with ampersands.

Example:

This example adds the `site` dimension and the `un` dimension to every tag. The `un` dimension is set to the variable *userName*. (The exact method for setting dimensions to variables depends on the scripting language used by your site.)

```
var NTPT_GLBLEXTRA = 'site=dailyherald&un=' + userName;
```

Basic page tags—local dimensions

If the dimension applies to some pages only, you must set the parameter for the dimension to the appropriate value on each page, using the NTPT_PGEXTRA variable. Separate name-value pairs with ampersands. Make sure NTPT_PGEXTRA is in a position where it will be parsed before the page tag script is called.

Example:

```
<script language="JavaScript">
var NTPT_PGEXTRA='author=JohnSmith';
</script>
```

Event tags

In the JavaScript event handler or Flash ActionScript for the event being tagged, do one of the following:

- Use ntptAddPair to define the dimension before the ntptEventTag, ntptLinkTag, or ntptSubmitTag function is called.
- Use the querymod argument of the ntptEventTag, ntptLinkTag, or ntptSubmitTag function to define the dimension.

Example:

In this example, the onclick JavaScript event handler calls the ntptEventTag function and uses the querymod argument to add the `ev` standard dimension and the `evdetail` custom dimension.

```
<a href="go.asp"
onclick="ntptEventTag('ev=buttonclick&evdetail=go');">
```

Direct tags

Include the name-value pair for the dimension in the list of name-value pairs after the question mark (?) in each direct tag to which the dimension applies.

Example:

This example adds the `site` dimension and the visitor identification cookie.

```
http://pt001.unica.com/ntpagetag.gif?lc=http%3A%2F%2Fapplication%2Fsta
rt&site=application&ck=UnicaNIODID%3DcookieValue
```

5 Including Cookie Data in Page Tag Requests

About including cookie data in page tag requests

The Unica NetInsight OnDemand page tag script automatically sets a visitor identification cookie and includes the cookie data in the page tag request. If your web site sets other cookies, you can configure the page tag script to include the data from those cookies in the page tag request as well.

About visitor identification cookies

By default, the Unica NetInsight OnDemand page tag script assigns a cookie to visitors. Unica NetInsight OnDemand uses this cookie for visitor identification and parameter analysis. The visitor identification cookie is set when the page tagging script loads.

Because the cookie is set through JavaScript, visitors must have JavaScript enabled in their browser in order to be tracked. (Users who do not have JavaScript enabled appear in the unique visitor count but no page tag data is collected for them.)

You can edit the page tag script (ntpagetag.js) to make the following changes to the visitor identification cookie:

- Stop setting visitor identification cookies

- Change the visitor identification cookie name (not recommended if you are already collecting data)
- Change the expiration period for the visitor identification cookie
- Declare a domain visitor identification cookie to track visitors across subdomains

To capture data from additional cookies

If desired, you can capture data from cookies set by your web site and then run reports on that data in Unica NetInsight OnDemand. You can capture cookie data globally (this is, on each page load) or for specific pages only. Cookie names are specified as an array.

To capture your cookie data via page tags:

1. In the page tagging script (ntpagetag.js), set NTPT_FLDS.ck to true. This enables collection of cookie data.

   Ensure that no variables used in your page tags at either the global or page level have their value set to `ck` as this would prevent collection of cookie data (for example, NTPT_GLBLEXTRA="ck=somevalue"). For the same reason, do not use tagging functions (for example, ntptAddPair, ntptEventTag) to add a `ck` parameter or modify the existing `ck` parameter.

2. Declare the cookie name variable. Specify the names of the cookies you want to capture as an array.

   a. To capture cookies globally, declare the NTPT_GLBLCOOKIES variable in the page tagging script.

   b. To capture cookies on specific pages, declare the NTPT_PGCOOKIES variable on those pages where you want to capture cookie data.

   Examples:

```
NTPT_GLBLCOOKIES= ["cookie1", "cookie2"];
NTPT_PGCOOKIES= ["cookie1", "cookie2"];
```

6 Tagging Events

What is an event?

An event is any on-page action other than loading a page on your site. Events include but are not limited to the following:

- Changing a field on a form
- Selecting an option in a drop-down list box
- Submitting a form
- Clicking a link to display a page on another site
- Starting a video
- Viewing a PDF document
- Checking an option (such as sound on/off)

Events are plentiful in RIAs such as Flash or AJAX applications.

The Unica page tag script includes special functions to use when tagging events.

About event tagging

Unlike a basic page tag, which sends a request when a page loads, an event tag sends a request when an event occurs on a page.

Another difference is that a basic page tag calls the entire page tagging script and an event tag calls a specific function in the script. There are three main event tagging functions:

- ntptEventTag
- ntptLinkTag
- ntptSubmitTag

Every event you want to track must call one of these functions in its JavaScript event handler or Flash ActionScript.

Events may also call ntptAddPair and ntptDropPair in conjunction with one of the main functions.

Counting events as events and not page views

Unica NetInsight OnDemand considers every call to the page tagging script a page view, even if the call is to one of the event tagging functions. To cause an event to count as an event and not a page view, set the field-value pair ev=*eventtype* using the query modifier of the ntptEventTag, ntptLinkTag, or ntptSubmitTag function.

Before you tag events

Unica NetInsight OnDemand analyzes events that occur on tagged pages only. A visitor must have at least one tagged page view in the visit before Unica NetInsight OnDemand tracks events for that visitor. A tagged event is tracked only if the previous page view for the visitor matches the page on which the event occurred. (If you are using direct tags, the `lc` parameter for the event must match the `lc` parameter for the previous page view.)

Tracking an event as a page view

Some events should be tracked as page views. For example, opening a PDF file is technically an event because clicking the link that opens the PDF is an interaction with a page. Opening a PDF can not be tracked with the basic page tag, since a basic tag must be embedded in an HTML page. However, usually what you want to analyze about the PDF view is the content, not the act of opening the PDF. Even though you must tag opening the PDF using the event tagging functions, you can have Unica NetInsight OnDemand track it as a page view instead instead of an event.

When to track an event as a page view

The main question to ask when deciding if something should be a page view or an event is this: do you want to analyze this content in path reports? If the answer is yes, Unica NetInsight OnDemand must track it as a page view.

How to track an event as a page view

To track an event as a page view, do the following:

- Do not set a value for the `ev` dimension.

  When the `ev` dimension is missing, the action does not count as an event.

- Either set the `pv` dimension to 1 (`pv=1`) or remove it altogether, but do not set `pv` to 0.

  When `pv=0`, the action does not count as a page view.

- Use the href as the value of the `lc` dimension, but be sure to use URL encoding for the value (of the href) so it does not conflict with the page tag request.

Example

```
<a href="article1.pdf" onclick="ntptEventTag('lc=' +
encodeURIComponent(this.href) );" target="_blank">
```

Ensuring link and submit page tag requests are submitted before the page unloads

Following a link or submitting a form ultimately leads to the unloading of the page that contains the link or the form. To ensure that the request is sent before the page is unloaded, the page tag script introduces a small delay when tagging links and submissions. This delay is the maximum amount of time that will elapse before the page is unloaded; if the request returns before the specified time has elapsed, the page will be unloaded immediately, without waiting for the rest of the time.

You can configure this delay globally by setting the NTPT_MAXTAGWAIT variable. You can adjust the wait for an individual link or submission by using the maxtagwait argument for ntptLinkTag or ntptSubmitTag.

The NTPT_MAXTAGWAIT variable does not introduce a delay for tags created using ntptEventTag. If you need a delay, use ntptLinkTag or ntptSubmitTag.

The default wait is one second, which is usually indiscernible to the user and is usually sufficient to effectively capture all tagged links and form submissions.

To tag JavaScript or AJAX events

1. Ensure that each page that contains an event you want to track has been tagged with the Unica page tag, either directly on the page or through a common include file.
2. Customize the page tagging as needed for each page, using the optional page-specific variables.
3. Create a JavaScript event handler for every event you want to track.

   Each event handler should call the appropriate event-tagging functions provided by Unica. (See the Reference chapter for details.)
4. Copy your tagged HTML pages to the Web server or your content management system.

The example below shows three different ways to call the Unica event-tagging functions. The first instance uses the onchange attribute of the <input> HTML element to call the ntptEventTag function directly. The second instance uses the onchange attribute of the <input> HTML element to call the ntptEventTag function and pass additional information. In this case, the value of the text box is passed to the page tag request. The final instance uses a custom event handler named MyEventHandler to call the ntptAddPair and ntptEventTag functions in a JavaScript function.

```
<form>
 <input type="checkbox" name="mybox" onchange="ntptEventTag();">
 <input type="text" name="mytext"
    onchange="ntptEventTag('ev="myevent"&mytext=" + encodeURIComponent(
this.value ) + "' );">

 <script language="JavaScript">
     function MyEventHandler() { ntptAddPair( "color", "red" );
ntptEventTag( "ev=myevent" ); }
 </script>
</form>
```

To tag events in Flash 8 or greater

1. Add the following function to the bottom of **ntpagetag.js**. Or place it in its own file and, in the page whose Flash you want to track, call the file after you call **ntpagetag.js**.

   unicaFunctions is a handler that parses name/value pairs returned by your tagged Flash and communicates them to the ntpagetag script

```
function unicaFunctions(command, args) {

    if (null == args) {
        return;
    }
    var tmpargs = args.split(",");

    if (command == "ntptEventTag") {
        (0 == tmpargs[0].length) ? ntptEventTag() :
ntptEventTag(tmpargs[0]);
    }
    else if (command == "ntptAddPair") {
        if ( 2 != tmpargs.length ) {
            return;
        }
        else {
            ntptAddPair(tmpargs[0], tmpargs[1]);
        }
    }

    else if (command == "ntptDropPair") {
        if ( 1 != tmpargs.length ) {
            return;
        }
        else {
            ntptDropPair(tmpargs[0]);
        }

    }
}
```

2. On the page whose Flash you want to track, add this function to your Flash ActionScript. It serves as a bridge to ExternalInterface.call which calls unicaFunctions. You can name the function anything.

```
function  FlashTracking(func, args) {
ExternalInterface.call("unicaFunctions", func, args);
}
```

3. In the page's ActionScript, use the function you just added to specify name/value pairs to pass to ntptEventTag. See the next section for examples.

Flash page tagging examples

In this example, the string "ev=videoaction&video=myvideo&videoaction=rewind" is passed to ntptEventTag. Unica NetInsight OnDemand reports will show that the visitor initiated a "rewind" "videoaction" event for "myvideo."

```
// visitor is rewinding movie
    this.FlashTracking( "ntptEventTag",
"ev=videoaction&video=myvideo&videoaction=rewind" );
```

Using ntptAddPair to add information

If you need to add information to an event's attributes prior to firing the event, you can call the Unica NetInsight OnDemand ntptAddPair helper function.

```
// Visitor sets interior color to red and adds the sunroof option
    this.FlashTracking( "ntptAddPair", "color,red" );
    this.FlashTracking( "ntptAddPair", "sunroof,true" );
    this.FlashTracking( "ntptEventTag", "ev=PersonalizeCar" );
```

This example will result in an event tag equivalent to:

```
  this.FlashTracking( "ntptEventTag",
"ev=PersonalizeCar&color=red&sunroof=true" );
```

Using ntptDropPair to remove information

While arguments specified in ntptEventTag are bound to just that call, name/value pairs set with ntptAddPair are stored in the JavaScript of the containing page. These pairs remain until a new page is viewed (which is effectively the length of the Flash application). To remove them use ntptDropPair. Building on the previous example, if the next set of ActionScript calls are:

```
 // Visitor removed the sunroof option and added ABS
    this.FlashTracking( "ntptDropPair", "sunroof" );
    this.FlashTracking( "ntptAddPair", "ABS,true" );
    this.FlashTracking( "ntptEventTag", "ev=PersonalizeCar" );
```

... this will result in an event tag equivalent to:

```
this.FlashTracking( "ntptEventTag",
"ev=PersonalizeCar&color=red&ABS=true" );
```

The "color=red" pair came from the original calls. The "sunroof=true" is no longer there because it was dropped by the ntptDropPair call. The "ABS=true" pair comes from the most recent ntptAddPair.

Tagging Flash to track links

In the following example a mouse down event on button_1 reports the event "clickedlink" to the external site unica.com.

```
button_1.addEventListener(MouseEvent.MOUSE_DOWN,mouseDownHandler);

function mouseDownHandler(event:MouseEvent):void {
    navigateToURL(new URLRequest("http://www.unica.com/"));
    this.FlashTracking("ntptAddPair", "gone,page");
    this.FlashTracking("ntptEventTag", "lc=http://www.unica.com/" +
"&ev=clickedlink");
}
```

Marking link tags as links to an external site

In Unica NetInsight OnDemand, links to external sites can be analyzed in the Link Summary. You can mark a link page tag request as a link to an external site by including the name-value pair `lk=1` in the page tag request’s query string. You can pass the `lk=1` name-value pair into the query string by including it in the querymod argument for ntptLinkTag or by using ntptAddPair to add it before you call ntptLinkTag.

When you use `lk=1`, clicking the link is not tracked as a page view or an event, and Unica NetInsight OnDemand does not store the page on which the visitor clicked the link.

Sending visit-level data when an event occurs

Sometimes you need to send information when an event occurs, but you do not want the information associated with the event in your reports. For example, when a visitor uses a specific search tool for the first time or becomes a customer (not just a visitor) for the first time, you want to associate those changes in status with the visit, not with a particular event.

You can send visit-level data by using an event tag with both of the following conditions:

- The tag does not contain `ev=` *event tag*.
- The tag contains `pv=0`.

The first condition keeps Unica NetInsight OnDemand from registering the tag as an event. The second keeps Unica NetInsight OnDemand from registering it as a page view.

7 Tagging Retail Activity

About retail activity

You can use page tags to track these retail activities:

- Product views
- Shopping cart adds
- Shopping cart removes
- Checkouts

You can tag or not tag individual retail activities, depending on your reporting needs. For example, if you require checkout data only, or if your site does not use a shopping cart, you can implement checkout tags without tagging cart adds, cart removes, or product views. Similarly, if your site is a product review site and not a retail site, you can use product view tags without tagging cart adds, cart removes, or checkouts.

Tagging product views

To track how often a product is viewed, use the `rtv` ("retail view") parameter to pass the product's identifier. You can pass the `rtv` parameter in a page tag (through NTPT_PGEXTRA) or in an event tag (through the querymod argument of ntptEventTag, ntptLinkTag, or ntptSubmitTag).

If a page contains multiple products, you can pass multiple values to the `rtv` parameter. Separate multiple values with semi-colons.

Example: Passing multiple static product identifiers

In this example several product identifiers are passed as static values:

```
var NTPT_PGEXTRA="rtv=prod1234;prod3456;prod5678";
```

Example: Passing a product identifier written by a script

In this example, the product SKU is written by a script (in this case, PHP):

```
var NTPT_PGEXTRA = "rtv=<?php echo $row_current_row['sku']; ?>";
```

Example: Appending rtv to an existing NTPT_PGEXTRA value

In this example, the `rtv` parameter is appended to the pre-existing value of NTPT_PGEXTRA:

```
NTPT_PGEXTRA +="rtv=prod1234;prod3456;prod5678";
```

Ensuring product views are not counted twice

If you pass the `rtv` parameter value through NTPT_PGEXTRA and the product view page contains event tags, you must clear the `rtv` parameter value before the event. Otherwise, two product views are logged: the first when the page loads and the second when the event tag is sent.

If NTPT_PGEXTRA passes only the `rtv` parameter, you can clear the value as follows:

```
NTPT_PGEXTRA="rtv="
```

However, typically NTPT_PGEXTRA passes additional information, such as authenticated user names, that you do not want to clear. In these cases, use the ntptDropPair function to clear the `rtv` parameter value without clearing the other parameters passed in NTPT_PGEXTRA. Call ntptDropPair before the event tagging function.

```
ntptDropPair(rtv);ntptEventTa('ev=cartadd&rta='sProduct + ';' +
iQuantity + ';' + fPrice);
```

Another method of ensuring product views are not counted twice is to store the original NTPT_PGEXTRA variable prior to calling the page tagging script (ntpagetag.js) and then reset the variable after the page tag image request is sent. This is particularly useful in situations where there might be multiple events on the product view page (and therefore multiple places where you would need to add ntptDropPair). The following example assumes that NTPT_PGEXTRA is already passing the authUser variable as the `un` parameter value (see the first line of the script):

```
<script language="JavaScript">
var NTPT_PGEXTRA = "un=" authUser;
var NTPT_PGEXTRA_ORIGINAL = NTPT_PGEXTRA;
NTPT_PGEXTRA +="rtv=prod1234;prod3456;prod5678";
</script>
<!--BEGIN: Unica Page Tag-->
<script language="JavaScript src="/scripts/ntpagetag.js"></script>
<!--END: Unica Page Tag-->
<script language="JavaScript">
NTPT_PGEXTRA=NTPT_PGEXTRA_ORIGINAL;
</script>
```

Tagging additions to a shopping cart

When a product is added to a shopping cart, pass its identifier, quantity, and unit price in that order through the `rta` ("retail add") parameter. You can pass the `rta` parameter in a page tag (through NTPT_PGEXTRA) or in an event tag (through the querymod argument of ntptEventTag, ntptLinkTag, or ntptSubmitTag).

```
ntptEventTag('ev=cartadd&rta='sProduct + ';' + iQuantity + ';' +
fPrice );
```

The product identifier, quantity, and price are all required in order for Unica NetInsight OnDemand to track the cart addition.

Example: In this example, the values for a single product are passed:

```
ntptEventTag('ev=cartadd&rta='prod1234 + ';' + 1 + ';' + 10.00 );
```

The unit price must be a float value. It must not include a currency sign or commas. Including a currency sign will result in a unit price of zero within Unica NetInsight OnDemand.

Information for multiple products can be passed in a single call. The first three values must represent the identifier, quantity, and unit price of the first product, the next three values the ID, quantity, and unit price of the second product, and so on.

Example: This example passes values for two products:

```
ntptEventTag('ev=cartadd&rta='prod1234 + ';' + 1 + ';' + 10.00 + ';'
+ 'prod5678 + ';' + 5 + ';' + 5.00 );
```

Tagging removals from a shopping cart

When a product is removed from a shopping cart, pass the same information that you sent when it was added. Pass the removed product's identifier, quantity, and unit price in that order through the `rtr` ("retail remove") parameter.

```
ntptEventTag('ev=cartremove&rtr='prod1234 + ';' + 1 + ';' + 10.00 );
```

As with adds, information for multiple products can be passed in a single call. The unit price must be a float value. It must not include a currency sign or commas.

Tagging the checkout process

Tagging the checkout process allows Unica NetInsight OnDemand to report which products a visitor purchased, the price and quantity at which they were purchased, and the total revenue for the order. You tag the checkout process by passing the appropriate parameters via the order confirmation (or similar) page served at the end of the checkout process.

You can pass the checkout parameters in a page tag (thought NTPT_PGEXTRA) or in an event tag (through the querymod argument of ntptEventTag, ntptLinkTag, or ntptSubmitTag). These are the checkout parameters:

- `rti` ("retail order number"): The order identification number. Including the rti parameter is optional. However, if it is left out Unica NetInsight OnDemand will not store the revenue for this checkout separately. Instead it will add it to the revenue of any other checkouts completed during the visit.
- `rtt` ("retail total revenue"): The total revenue for this order.
- `rtc` ("retail checkout"): The product ID, quantity, and unit price of each product purchased.

Syntax:

```
var NTPT_PGEXTRA =
"rti=sOrderID&rtt=fTotalRevenue&rtc=sProduct;iQuantity;fPrice" ;
```

Example: This example shows an order ABC1234 that totals $199.99 and consists of two products costing $150 and $49.99 respectively.

```
var NTPT_PGEXTRA =
"rti=ABC1234&rtt=199.99&rtc=prod1234;1;150;prod3456;1;49.99" ;
```

Retail page tag parameters summary

Page Tag Value	What It Reports	Format / Example

rtv	retail (product) view	rtv=sProduct
		NTPT_PGEXTRA="rtv=product1;product2;product3";

rta	retail (cart) add	rta=sProduct;iQuantity;fPrice
		ntptEventTag('ev=cartadd&rta='prod1234 + ';' + 1 + ';' + 10.00);

rtr	retail (cart) remove	rtr=sProduct;iQuantity;fPrice
		ntptEventTag('ev=cartremove&rtr='prod1234 + ';' + 1 + ';' + 10.00);

rtc	retail (cart) checkout	rtc=sProduct1;iQuantity1;fPrice1;sProduct2;iQuantity2;fPrice2
		NTPT_PGEXTRA="rtc=prod1234;1;5.00;prod3456;3;19.95";

rtt	retail total revenue	rtt=float
		rtt=210.54

rti	retail order number	rti=string
		rti=8318

vc	visit cost	vc=float
		vc=20.00

Retail metrics

This table lists the retail metrics that are available in Unica NetInsight OnDemand reports and provides the corresponding parameter and a description for each one.

Metric	Parameter	Description
Abandoned Carts	rta	The number of visits that have shopping carts that did not complete the checkout process.
Abandoned Revenue	rta	The total value of all shopping carts that did not complete the checkout process.

Metric	Parameter	Description
Average Order Value	rtt	The amount of money (on average) spent on each order.
Cart Abandonment Rate	rta	The percentage of visits that have shopping carts that did not complete the checkout process.
Cart Adds	rta	The number of times visitors placed a product in their shopping carts.
Cart Removes	rtr	The number of times visitors deleted a product from their shopping carts.
Checkout Rate	rtc/rtt	The percentage of visits with shopping cart activity that completed the checkout process.
Checkouts	rtc/rtt	The number of times the checkout process was completed.
Product Revenue	rtc	The total revenue associated with products.
Product Views	rtv	The number of times visitors viewed a product.
Quantity Abandoned	rta	The total number of units in abandoned carts.
Quantity Added	rta	The total number of units added to a cart.
Quantity Purchased	rtc	The total number of units purchased.
Quantity Removed	rtr	The total number of units removed from a cart.
Revenue	rtt	The total order revenue. May include revenue (such as shipping charges and taxes) not associated with a product.

Differences between Product Revenue and Revenue

The Product Revenue metric is the result of multiplying the quantity for each purchased product (passed by the `rtc` parameter) by the price for each product (also passed by the `rtc` parameter).

The Revenue metric is the value passed by the `rtt` parameter.

If your order totals (passed by the `rtt` parameter) include shipping charges, taxes, or other fees that are not defined as products, the Revenue metric total and Product Revenue metric total will be different.

If you want to report on shipping charges, taxes, or other fees, Unica recommends that you add those charges to the checkout tag as products passed by the `rtc` parameter.

8 Tagging Applications and Widgets

About tagging applications and widgets

In order to track an application or widget using basic page tags, the following conditions must both be true:

- The application or widget must support JavaScript.
- You can place the page tag script on the application or widget pages.

If you cannot use basic page tags, you can use direct tags to send usage data from the application or widget to Unica NetInsight OnDemand.

Direct tags and page tags are built using the same set of page tag parameters. Like page tags, direct tags can contain data for custom dimensions.

Direct tags (sometimes called web beacons) bypass the page tag script (ntpagetag.js) and request the page tag image directly from the server. Because direct tags bypass the script, there is no default direct tag. You must define in the direct tag each item of information you want Unica NetInsight OnDemand to receive.

Unlike page tags, direct tags work in situations where JavaScript rendering is not possible.

Anatomy of a direct tag

There are two parts to a direct tag:

- The URL for the page tag image on the Unica NetInsight OnDemand server. This is the same for every tag you set.
- A query string that contains the information you want sent to Unica NetInsight OnDemand.

Simple Example

This direct tag requests pt001.unica.com/ntpagetag.gif and passes information using the `lc`, `site`, and `ck` parameters:

```
http://pt001.unica.com/ntpagetag.gif?lc=http%3A%2F%2Fapplication%2Fsta
rt&site=application&ck=UnicaNIODID%3DcookieValue
```

The request shows up in Unica NetInsight OnDemand as a page view of http://application/start, and the page view is included in any profile that contains `site=application`.

Example with Custom Dimensions

This direct tag passes information using the `lc`, `site`, `ck`, and `un` parameters and the custom parameters version and layout:

```
http://pt001.unica.com/ntpagetag.gif?lc=http%3A%2F%2Fapplication.compa
ny.com%2Fproducts%2Fproduct001&site=application&ck=UnicaNIODID%3Dcooki
eValue&un=UnicaUser&version=1.0&layout=3A
```

Best practices for direct tags

- Thoroughly test your direct tags in a data validation profile to ensure they are sending the data you need. Because you build direct tags from the ground up, testing is especially important.
- Include the `site` parameter in all direct tags.
- Include at least one visitor identification parameter.
- If the application or widget might be placed on a secure page or site, reference the page tag via the secure URL.

About the lc parameter

The value of the `lc` parameter indicates the page being viewed or the page on which the event occurred. The value must be in the form of a URL (for example, http://application) and must be encoded.

The `lc` parameter should include a descriptive page name, including as much detail as possible about what is going on within the application at the time the page tag image is requested. Typically, content categorization can be derived directly from the value of the `lc` parameter.

For example, the following image request tells us that the visitor was in the products section of the application because the `lc` parameter value uses a directory structure to indication content hierarchy:

```
http://pt001.unica.com/ntpagetag.gif?lc=http%3A%2F%2Fapplication%2Fpro
ducts%2F&site=application
```

When NOT to pass the lc parameter

You can leave the `lc` parameter out of the direct tag if both of the following conditions are true:

- You are tagging a web-based application, widget, or viral video.
- Your primary concern is the sites on which your content is installed and how many times it is loaded on any given page or site.

When the `lc` parameter is missing, Unica NetInsight OnDemand reads the referring URL from the headers of the request and treats the referring URL as the page being viewed.

If you must track events or interactions within the application, pass the `lc` parameter and parse the referring URL as a parameter.

Direct tags sent from applications that are not web-based must contain an `lc` parameter.

Where to place your direct tags

The best location for a direct tag depends on the application or widget you are tagging. In general, you can place a direct tag in any location where you can place an image request.

Minimizing the number of tags

You should not tag events unnecessarily. If an action results in loading a new page and you can pass the information about the event in the tag for the page view, you may not need to tag the event.

For example, if a user clicked a Hot Products article link from an application's start page, the tag for the subsequent product detail page could pass a click referral parameter (`clickref`) that specifies which link was clicked. The tag would look like the following:

```
http://pt001.unica.com/ntpagetag.gif?lc=http%3A%2F%2Fapplication.compa
ny.com%2Fproducts%2Fproduct001&site=application&ck=UnicaNIODID%3D81759
178198560&un=UnicaUser&version=1.0.2&layout=3A&clickref=start-
hotproducts
```

9 Reference

Introduction

This section provides reference information on the functions, variables, and fields available through page tags.

Required variables

About required variables

Required page tagging variables are defined in the page tag script (ntpagetag.js).

NTPT_IMGSRC

A string variable that contains the URL of the page tag image. The URL must point to the Unica OnDemand server and is provided by your Unica Technical Account Manager.

Syntax

```
var NTPT_IMGSRC = ‘URL’;
```

Example (without protocol and server)

```
var NTPT_IMGSRC = '/images/ntpagetag.gif';
```

Example (fully qualified)

```
var NTPT_IMGSRC = 'http://mysite.com/images/ntpagetag.gif';
```

NTPT_FIELDS

An object variable that specifies the fields that will be included in the default page tag. Fields set to true will be included in the page tag. Fields set to false will not be included. In most cases, you should use the default field settings.

Every page tag should include the lc field.

Syntax

```
var NTPT_FLDS = new Object();
NTPT_FLDS.lc = [true,false]; // Document location
NTPT_FLDS.rf = [true,false]; // Document referrer
NTPT_FLDS.rs = [true,false]; // User's screen resolution
NTPT_FLDS.cd = [true,false]; // User's color depth
NTPT_FLDS.ln = [true,false]; // Browser language
NTPT_FLDS.tz = [true,false]; // User's timezone
NTPT_FLDS.jv = [true,false]; // Browser's Java support
NTPT_FLDS.ck = [true,false]; // Allow capture of cookie values
```

Example

```
var NTPT_FLDS = new Object();
NTPT_FLDS.lc = true; // Document location
NTPT_FLDS.rf = true; // Document referrer
NTPT_FLDS.rs = true; // User's screen resolution
NTPT_FLDS.cd = true; // User's color depth
NTPT_FLDS.ln = true; // Browser language
NTPT_FLDS.tz = true; // User's timezone
NTPT_FLDS.jv = true; // Browser's Java support
NTPT_FLDS.ck = true; // Allow capture of cookie values
```

NTPT_MAXTAGWAIT

The maximum number of seconds that a call to ntptLinkTag() or ntptSubmitTag() will wait before following the link or submitting the form. The delay is intended to ensure that the page tag request is sent before the tagged page is unloaded. The event will be executed when the page tag request is returned or the specified number of seconds has elapsed, whichever happens first.

Syntax

```
var NTPT_MAXTAGWAIT = number;
```

Possible values

Value	Description
Any number greater than 0	The number of seconds to wait before following the link or form submission
-1	Execute the event immediately, without waiting

Examples

```
var NTPT_MAXTAGWAIT = 1; // one second
var NTPT_MAXTAGWAIT = 2.5; // two and a half seconds
var NTPT_MAXTAGWAIT = 0.1; // a tenth of a second
var NTPT_MAXTAGWAIT = -1; // no delay
```

Optional global variables

About optional global variables

Optional global variables can be specified in the page tag script (ntpagetag.js).

NTPT_HTTPSIMGSRC

The URL of the page tag image to use when the tagged page is accessed using the “https:” protocol.

Syntax

```
var NTPT_HTTPSIMGSRC = 'image';
```

Example

```
var NTPT_HTTPSIMGSRC = 'https://mysite.com/images/ntpagetag.gif';
```

NTPT_GLBLEXTRA

A query modifier that will be applied to every page tag and event tag. The query modifier should contain the key-value pairs you want to add to or delete from the query string for your tags.

Syntax

```
var NTPT_GLBLEXTRA = 'key=value';
```

Example

```
// Append the 'sitetheme=blue' pair to every page tag.
var NTPT_GLBLEXTRA = 'sitetheme=blue';
```

NTPT_GLBLREFTOP

Retrieve the referrer (the value for the “rf” field) from the top (that is, the most containing) frame of the current page. Otherwise, the referrer is retrieved from the current page. If you do not set this variable, it defaults to false.

Syntax

```
var NTPT_GLBLREFTOP = [true,false];
```

Example

```
var NTPT_GLBLREFTOP = true;
```

NTPT_GLBLCOOKIES

An array of customer-set cookies set globally to be captured by the page tagging script.

Syntax

```
var NTPT_GLBLCOOKIES = [ ];
```

Example

```
var NTPT_GLBLCOOKIES = ["cookie1", "cookie2"];
```

NTPT_SET_IDCOOKIE

Tells the page tagging script to set the visitor identification cookie.

Syntax

```
var NTPT_SET_IDCOOKIE = [true, false];
```

Example

```
var NTPT_SET_IDCOOKIE = true;
```

NTPT_IDCOOKIE_NAME

Sets the name of the visitor identification cookie.

Syntax

```
var NTPT_SET_IDCOOKIE_NAME = "";
```

Example

```
var NTPT_SET_IDCOOKIE_NAME = "unique_visitor";
```

NTPT_IDCOOKIE_EXPIRE

The expiration time (in seconds) of the visitor identification cookie. If not specified the value defaults to 155520000 (60 months).

Syntax

```
var NTPT_IDCOOKIE_EXPIRE = number;
```

Possible values

Value	Description
Any number	The number of seconds until the cookie expires

Examples

```
var NTPT_IDCOOKIE_EXPIRE = 315360000; // 10 years
```

NTPT_IDCOOKIE_DOMAIN

Specifies the domain suffix for the visitor identification cookie. This allows customers with more than one site within the same domain to use a single version of the visitor identifaction cookie, allowing you to track cross-site visits/visitors. For example, if your site uses the subdomains www.mydomain.com, info.mydomain.com, and support.mydomain.com, you can ensure they use the same identification cookie by specifying an NTPT_IDCOOKIE_DOMAIN value of ".mydomain.com"

Syntax

```
var NTPT_IDCOOKIE_DOMAIN = "";
```

- The domain suffix specified as the value must include at least two periods (see example).
- The domain suffix must be the same domain that sends the cookie. For example, you may not set a cookie for abc.com if your server's domain is mydomain.com.

Example

```
var NTPT_IDCOOKIE_DOMAIN = ".mydomain.com";
```

Page-specific variables

About page-specific variables

Page-specific variables are optional variables that you can use to override the default page tag settings for the page on which the page-specific variables are used. The variables are not included in the page tag script (ntpagetag.js).

Where to put page-specific variables

Page-specific variables must be defined in the page to which you want them applied. They should be defined before the Unica page tag is loaded so they can modify the page tag request before it is sent. Because the <head> element for the page is processed first, it is often a good place to define the page-specific variables.

NTPT_PGEXTRA

A list of field-value pairs that will be applied to the page tag for the current page. To delete a field from the page tag, set it to an empty value. Although you can add any field-value pair, some fields are recognized automatically by Unica NetInsight and do not require you to create a parameter in order for them to appear in Unica NetInsight reports.

Syntax

```
var NTPT_PGEXTRA = 'field=value&field=value&field=value[...]';
```

Example

```
// Append the 'pagetheme=red' pair. Drop the 'rf' field.
var NTPT_PGEXTRA = 'pagetheme=blue&rf=';
// Treat this page as a 404 (Not Found) error.
var NTPT_PGEXTRA = 'sc=404';
```

NTPT_PGREFTOP

Retrieve the referrer (the value for the “rf” field) from the top (that is, the most containing) frame of the current page. Otherwise, the referrer is retrieved from the current page.

Syntax

```
var NTPT_PGREFTOP = [true,false];
```

Example

```
var NTPT_PGREFTOP = true;
```

NTPT_NOINITIALTAG

This variable can be set at both the global and page level. Its default is false. When set to true at the global level, it stops the page tag script from collecting initial page load data for all pages. When set to true at the page level, it stops the page tag script from collecting initial page load data for the current page. Event tags for pages will still be sent.

Syntax

```
var NTPT_NOINITIALTAG = [true,false];
```

Example

```
var NTPT_NOINITIALTAG = false;
```

NTPT_PGCOOKIES

An array of customer-set cookies for a specific page to be captured by the page tagging script.

Syntax

```
var NTPT_PGCOOKIES = [ ];
```

Example

```
var NTPT_PGCOOKIES = ["cookie1", "cookie2"];
```

Event tagging functions

- ntptAddPair
- ntptDropPair
- ntptEventTag
- ntptLinkTag
- ntptSubmitTag

ntptAddPair

Add the specified key-value pair to the query string of the next event tag to fire. If the key already exists in the query string, the value will be replaced. If the value is empty (‘’), the key will be dropped.

Syntax

```
ntptAddPair( key, value )
```

Argument	Description
key	The name of a name-value pair to add to the next event tag to fire.
value	The value of a name-value pair to add to the next event tag to fire.

Example

```
ntptAddPair( "color", "red" );
```

ntptDropPair

Drop the specified key-value pair from the query string of the next event tag to fire.

Syntax

```
ntptDropPair ( key )
```

Argument	Description
key	The name of a name-value pair to drop from the next event tag to fire.

Example

```
ntptDropPair( tmpargs[0] );
```

ntptEventTag

Fires an event tag using the working query string. This function should be called from a document element’s event handler.

Syntax

```
ntptEventTag ( [querymod] )
```

Argument	Description
querymod	A query modifier for the event tag. It modifies the working query string for the page tag.

Example

```
ntptEventTag( "ev=pickcolor" );
```

ntptLinkTag

Tags a link that would otherwise not be accessible to page tagging. These links include downloads, non-HTML pages, and pages on other Web sites. This function must be called from the onclick attribute of a link and should return the value of the function to the onclick handler.

☼ Link tags are not modified by the NTPT_PGEXTRA variable.

Syntax

```
ntptLinkTag ( linkobj [, querymod [, maxtagwait]] )
```

Argument	Description
linkobj	A link object. The keyword "this" tells the function to follow the link after tagging it.
querymod	A query modifier for the link tag. It modifies the working query string for the page tag.
maxtagwait	The maximum number of seconds that the call will wait before following the link. This overrides the global wait time specified by NTPT_MAXTAGWAIT.

Example

```
onclick="return ntptLinkTag( this );"
```

ntptSubmitTag

Tags form submissions. This function must be called from the onsubmit attribute of a form and must return the value of the function to the onsubmit handler.

Syntax

```
ntptSubmitTag( formobj [, querymod [, maxtagwait]] )
```

Argument	Description
formobj	A form object. The keyword *this* tells the function to submit the form after tagging it.
querymod	A query modifier for the submit tag. It modifies the working query string for the page tag.
maxtagwait	The maximum number of seconds that the call will wait before submitting the form. This overrides the global wait time specified by NTPT_MAXTAGWAIT.

Example

```
return ntptSubmitTag( document.myform );
```