

Tracking the use of mobile, web-based and other applications for analysis in NetInsight On Demand

This is a technical document explaining how the usage of various applications (e.g. iPhone applications, Flash widgets) may be tracked in Unica NetInsight by embedding requests for the Unica NetInsight page tag image at key points within the application. Other recommended documents include: *NetInsight On Demand Page Tagging Guide*, *NetInsight On Demand – Requesting Parameters*, *NetInsight On Demand – Event Tagging*.

How page tagging works

NetInsight tracks visitor activity by parsing page tag image requests. In a typical configuration, data is collected by including the Unica NetInsight page tag JavaScript on all pages of a site. The JavaScript gathers up some information as the pages load, and the data is sent in query string parameters of an image request. The image requests go to Unica's data collectors. For example:

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

To modify the data NetInsight receives when parsing these requests, page tag requests can be manipulated through JavaScript. The page tag image request itself can be mimicked in situations where JavaScript rendering is not possible. Embedded images, or “web beacons”, can provide an abundance of data to NetInsight, and this is the key to application tagging.

First step: Reporting requirements

In order to successfully plan a web beacon implementation, we must understand:

1. When web beacons will be called (at what points during use of the application)
2. What data will be passed to NetInsight at those times

In order to plan for these, we must first understand the reporting requirements and set some implementation goals. It is highly recommended that these requirements and goals be reviewed with Unica Professional Services or Technical Account Management. If Professional Services will not be engaged, application tagging should be pushed through iterative rounds of testing in a data validation profile as a cooperative effort with the application users, technical team, and Unica Technical Account Management to ensure that reporting requirements are met.

For the purpose of this document, let's proceed with a basic example where as a first phase of implementation, we need to establish a baseline and understand general navigation of an application. Our key metrics will be: # of visitors, # of users. Our key dimensions: versions in use, pathing, content categories, user demographic data.

Getting to know the Unica NetInsight page tag image request

Here is an example page tag image request as fired from the standard Unica On Demand page tag JavaScript (note that the domain the page tag image is hosted at will differ for your implementation). This example represents a page view of <http://blog.company.com/page.html>:

```
http://pt001.unica.com/ntpametag.gif?js=1&ts=1245238101072.441&lc=http%3A%2F%2Fblog.company.com%2Fpage.html&rf=http%3A%2F%2Fwww.google.com%2Fsearch&rs=1280x800&cd=32&ln=en&tz=GMT%20%2B04%3A00&jv=1&ck=UnicaNIOIDID%3DtIO5RwsKyD6-V5OcKhL%3BSessionID%3D9816106-8172&site=blog
```

Most of these name/value pairs, or query string parameters, you see here are not required when creating a web beacon request. Here's a breakdown of the information they provide to NetInsight:

Name	Value	Example
js	Set to zero for visitor that do not have JavaScript enabled.	js=1
ts	A unique floating point identifier used to help avoid cached page tag requests	ts=1245238101072.441
lc	URL of page (including the query string) or event requested. * The value must be escaped before passing	lc=http%3A%2F%2Fblog.company.com%2Fpage.html
rf	Referring URL/site (when present) to the requested page. * The value must be escaped before passing	rf=http%3A%2F%2Fwww.google.com%2Fsearch
rs	Web browser screen resolution (width x height)	rs=1280x800
cd	The visitor's web browser color depth	cd=32
ln	The visitor's web browser language	ln=en
tz	The visitor's web browser time zone * The value must be escaped before passing	tz=GMT%20-04%3A00
jv	Set to zero for visitor that do not have Java installed.	jv=1
ck	The visitor identification cookie, and any cookies passed through the NTPPT_GLBLCOOKIES variable. Contains semi-colon delimited, escaped name/value pairs as cookie=value. * The value must be escaped before passing	ck=UnicaNIOIDID%3DtIO5RwsKyD6-V5OcKhL%3BSessionID%3D9816106-8172
site	In NetInsight On Demand, the site= parameter is used to determine which data will be analyzed in each profile. A profile may include one or multiple site= parameter values for analysis.	site=blog

When creating beacon requests for an application, the only name/value pairs that are typically required are the lc= and site= parameters. The lc= parameter indicates page location, and site= is used by NetInsight to filter data into profiles for analysis. A profile may include one or multiple site= values. Generating a request for the following image:

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

would trigger a page view in NetInsight of http://application/start, and this data would be available in any profile including site=application.

Note: The lc= (location) field of the page tag image request must contain a URL (e.g. http://something). The lc= parameter and all other parameters must be escaped (URL encoded) if they might contain special URL characters (e.g. ? &).

The lc= field should include a descriptive page name, including as much detail as possible about what is going on within the application at the time the web beacon is requested. Content categorization was noted earlier in our example as a reporting requirement, and typically content categorization can be derived directly from the “URL” passed in the lc= field. For example, the following image request tells us that the visitor was in the products section of the application because the lc= field uses a directory structure to indicate content hierarchy:

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

When to NOT pass the lc= parameter

If the lc= parameter is not sent to NetInsight, NetInsight will read the referring URL from the headers of the request and treat the referring URL as the page being viewed. For distributed web based applications and viral video, this provides a way to determine what sites your content is installed on, and how many times it is loaded on any given page or site. Often, the initial tag fired from such an application / video / widget will not contain an lc= parameter. This does not work in non-web situations – web beacons being sent from applications that are not web-based must contain an lc= parameter.

What else should be passed to NetInsight?

There are a couple of best practices, but for the most part this depends on the application and business specific reporting requirements. There is a lot more that can be done!

Aside from the lc= and site= parameters, it is typically recommended to include one or two forms of visitor identification parameters. At the least, some form of anonymous user ID, if passed through the image request can provide a much higher level of accuracy in analysis than NetInsight’s default methodology when no such data is provided. If no visitor identification parameters (for example, the ck= parameter from the table above includes a UnicaNIDID cookie value) are passed to NetInsight, NetInsight will identify the visitor by a concatenation of IP address and user-agent (browser definition).

If the ability to pass an anonymous ID, GUID, mobile ID, or some other unique identifier exists, mimic the UnicaNIOIDID cookie seen in the ck= parameter. For example, if our application assigned UID 81759178198560 to a particular installation, it might include a ck= parameter as in:

```
http://pt001.unica.com/ntpametag.gif?lc=http%3A%2F%2Fapplication%2Fstart&site=application&ck=UnicaNIOIDID%3D81759178198560
```

This would result in NetInsight recognizing this particular user anonymously as '81759178198560', and grouping page views together into visits whenever that parameter is passed. This type of anonymous identification significantly improves the accuracy of unique visitor metrics, session (visit) metrics, and new and repeat visitor metrics.

In situations where authentication exists, a un= parameter containing a user ID or user name should be passed as well:

```
http://pt001.unica.com/ntpametag.gif?lc=http%3A%2F%2Fapplication.company.com%2Fstart&site=application&ck=UnicaNIOIDID%3D81759178198560&un=8105
```

Additional page and application level data can also be passed to NetInsight. In the next image request example, the application version and layout selection options are passed to NetInsight for further segmentation:

```
http://pt001.unica.com/ntpametag.gif?lc=http%3A%2F%2Fapplication.company.com%2Fproducts%2Fproduct001&site=application&ck=UnicaNIOIDID%3D81759178198560&un=UnicaUser&version=1.0.2&layout=3A
```

Events, page views, and visit-level data

All of the above requests would trigger a page view in NetInsight. The Page dimension is centric to page views, and pages are the building blocks for path analysis. Any interaction that should be clearly identified in path analysis should be tracked as a page view, and the page name (lc= parameter) should be descriptive enough to make the navigation clear.

NetInsight also gives us the ability to track interactions at a more detailed level than page view: the event level. Events are good for tracking on-page interactions in the web world (for example, interactions with a Flash application). In the application world, they are typically used to provide more detail around what the user is doing with the application (e.g. what exactly was clicked on). Events also come up frequently with application interactions that do not translate into much in the way of a change in content (for example, checking a sound on/off toggle).

If an interaction is to be tracked, and it is possible for it to be tracked during a "page view" request that is already being sent, we should take care to not fire any events (web beacons containing ev= parameters) unnecessarily. Clicking on a button does not need to trigger an event beacon if the application could just as easily pass the same information when the subsequent page is loaded. For example, if a "hot products" article link from application start were clicked, the subsequent page (product001's detail page - /products/product001) could pass a click referral parameter (clickref):

`http://pt001.unica.com/ntpametg.gif?lc=http%3A%2F%2Fapplication.company.com%2Fproducts%2Fproduct001&site=application&ck=UnicaNIOIDID%3D81759178198560&un=UnicaUser&version=1.0.2&layout=3A&clickref=start-hotproducts`

If within the application, it is not possible to pass the click data on to NetInsight from the subsequent “page view”, we could instead call the beacon when the click occurs. Adding `ev=eventtype` will prevent the click from being tracked as a page view. In this case, we pass `ev=newsclick` to indicate that someone clicked on a link within an article (as opposed to a navigation component), and pass `evdetail=hotproducts` to indicate what type of article was clicked.

`http://pt001.unica.com/ntpametg.gif?lc=http%3A%2F%2Fapplication.company.com%2Fstart&site=application&ck=UnicaNIOIDID%3D81759178198560&un=UnicaUser&version=1.0.2&layout=3A&ev=storyclick&evdetail=hotproducts`

Two requests fire in this scenario: one on the click (the event request is above), and another on the subsequent page view (the page view request is below):

`http://pt001.unica.com/ntpametg.gif?lc=http%3A%2F%2Fapplication.company.com%2Fproducts%2Fproduct001&site=application&ck=UnicaNIOIDID%3D81759178198560&un=UnicaUser&version=1.0.2&layout=3A`

Notice that when the event fires, the `lc=` parameter for the event matches the page view the event occurred on. NetInsight will not track an event if the previous page view for that visitor was of a different page, and a previous page view must exist.

Back to those reporting requirements

With all of the information reviewed above, have we put together enough tagging recommendations to allow our application users to create the reports they requested for phase one? Our key metrics were: # of visitors, # of users. Our key dimensions: version, pathing, content categories, user demographic data.

Number of Visitors - NetInsight will be able to calculate a # of visitors metric regardless of whether we send a `ck=` parameter to more accurately identify an application installation. Without a parameter, the user's IP address and the user-agent from the headers of the page tag image request will be used to identify the visitor and group page views into visits.

Number of Users – We'll be able to find out how many unique logged in users the application is seeing across any tracked timeframe thanks to the `un=` parameter.

Pathing – NetInsight will automatically string together page views into paths that can be analyzed in reporting. The `rf=` field is not required for path analysis.

Content Categories – Our descriptive naming convention for pages will allow content categorization to be parsed directly from the `lc=` parameter.

User Demographic Data – We aren't sending any user demographic data to Unica through the application tags, but we are sending user IDs. We'll send a data feed to categorize our users by various demographics for reporting.

What else can we find out?

Of course there are loads of default dimensions and metrics that NetInsight will provide, most of which will automatically work without additional customization. These include:

Metrics:

- # of views
- # of events
- # of visits
- # of new visitors
- # of repeat visitors
- Total Time Online
- Average Viewing Time
- Average Visit Duration
- Views per Visit
- Custom Metrics (filtered metrics, or multi-input metrics)

Dimensions:

- Browser
- Platform
- Entry Page
- Geographic Data (e.g. Country, City, Time Zone, Organization, etc.)
- Date
- Time
- Day of the Week
- Custom Dimensions (parameters parsed from page tag requests, or from data integrations)

Are there other page tag parameters NetInsight recognizes automatically?

Yes! Here's a list of other parameters (aside from the table on page 2) NetInsight automatically sets or parses:

Name	Value	Example
pv	Set to zero to avoid counting hit as a page view. Useful for sending visit-based data to NetInsight.	pv=0
ev	Set to any value to count request as a specific event type instead of a page view. Key concept for event tagging and RIA tracking.	ev=menuexpand
ets	A unique floating point identifier used to help avoid cached page tag requests	ets=1241461524461.967
un	Set to populate NetInsight "user" or "account id" dimension	un=8105

sc	Set to HTTP status code to change recorded status of page. Useful for recording page-not-found errors by tagging error page with sc=404.	sc=404
rtv	Semi-colon delimited list of product SKUs viewed on the page	rtv=FY210;EP360;...
rta	Semi-colon delimited list of product SKUs, quantities, and prices added to a shopping cart	rta=FY210;4;4.95;...
rtr	Semi-colon delimited list of product SKUs, quantities, and prices removed from a shopping cart	rtr=FY210;4;4.95;...
rtc	Semi-colon delimited list of product SKUs, quantities, and prices purchased in a shopping cart	rtc=FY210;4;4.95;...
rtt	Total value of products purchased in a shopping cart	rtt=19.80
rti	Unique order number for products purchased in a shopping cart	rti=8318
lk	Set to one (1) on links to external sites to track in the Link Summary.	lk=1

Conclusion

This concludes the technical overview of application tagging.