IBM Software Support for
Enterprise Marketing Management
Coremetrics Implementation

Implementation Guide

June 2011

http://support.coremetrics.com
cm_support@us.ibm.com

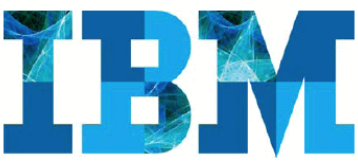

Table of Contents

Introduction

This document provides detailed instructions and best practices supporting implementation of Coremetrics tagging. It includes both a business level description of implementation requirements useful for Coremetrics stakeholders in your organization, as well as technical documentation for reference by developers. The Implementation Guide should be used in conjunction with any other documentation provided by Coremetrics. Please contact Coremetrics Support with any additional questions regarding Coremetrics implementation.

1 Technical Overview

1.1 *Tracking Technology*

Coremetrics collects data at the client browser level. Data is captured when a 'tagged' site page is rendered by the visitor's browser. JavaScript function calls, or 'tags', are executed as the requested page renders in the visitor browser, creating standard http GET image requests appended with query string parameter 'name=value' pairs obtained from the 'tag' parameter data strings. The requests are sent to Coremetrics data collection servers where the appended name=value pairs are extracted and loaded into your analytics data warehouse for subsequent reporting as actionable metrics.

The data parameters in the 'tag' function calls communicate information about pages, products and visitor activities, such as carting or selecting specific products, completing purchases, registering or logging on, interacting with specific page elements, etc.

- JavaScript tags function call format example:

```
<script type="text/JavaScript">
        cmTagName(<parameter_1>,<parameter 2>,<parameter 3>, etc);
</script>
```

The image request received by Coremetrics data acquisition servers is parsed to extract the visitor data from the name=value query string parameter pairs, loaded into the analytics data warehouse, and a 1 x 1 pixel GIF file is returned to the requesting browser. The data collection image request are made in memory, rather than written directly onto the page, preventing response 'images' from visibly rendering on the page. The image request is asynchronous with the page load and cannot interrupt page load or visitor experience. In the unlikely event that the request cannot reach the Coremetrics data collection servers or the request response pixel is not received by the requesting browser, the page continues to render: site functionality and visitor experience are unaffected.

1.2 *Cookies*

In order to facilitate tracking of session and visitor activities, Coremetrics makes use of cookies.

- **Session Cookie:** The 'Session' cookie exists only for the lifetime of the current browser session. The Session cookie will exist from the point at which the first tag is received from the website until 1) the visitor closes all browser windows for the browser in question or 2) more than 30 minutes pass without receiving a data collection tag from the browser session. One or more 'sessions' may be associated with a 'visitor'.

- **Visitor Cookie:** The 'Visitor' cookie persists after the visitor closes all browser windows. The 'Visitor' cookie contains a cookie ID referenced by Coremetrics to identify a visitor returning to the site across multiple 'sessions'.

In addition to the 'Visitor' and 'Session' cookies, several additional session-based cookies may be set depending on your specific implementation and Coremetrics version. Certain optional Coremetrics applications may set additional session cookies (Intelligent Offer, AdTarget and others). See Section 8.7 'Coremetrics Cookes – Technical Appendix' for more details.

1.3 *Secure Protocols*

Coremetrics can make image requests in either HTTP or HTTPS protocols. If the page on which the image request is being made is a secure page using the HTTPS protocol, the image request will be made through HTTPS.

A subset of tag types defaults to HTTPS to ensure secure transmission of the data: Registration and Order tag. Additional tags can be specified for default HTTPS if desired. Consult Coremetrics Support if any changes are desired.

1.4 *Coremetrics Service Domains*

To insure that Coremetrics tools, reporting and data collection function correctly within your internal corporate network, please allow the following domains inbound and outbound access to your network:

- testdata.coremetrics.com
- test.coremetrics.com
- data.coremetrics.com
- welcome.coremetrics.com
- itt.coremetrics.com
- Any Coremetrics-Managed data collection domains in use (also see section 6).

Coremetrics may change the IP's associated with these domains without notice, so it is not possible to reliably allow access to these domains by IP.

In addition, browsers used by development, QA and other internal network teams should allow:

- JavaScript execution
- Cookies set by the domains listed above (testdata.coremetrics.com, etc.)

1.5 *Multibyte Characters*

Multibyte characters collected in tag data sent to a singlebyte Coremetrics Client ID will appear in reporting as '?' and '□' characters. This commonly occurs when multibyte Trademark (tm) or Registration symbols are included in otherwise singlebyte data. Other symbols such as '-' (dash) have multibyte and singlebyte versions. Where possible, collect the single byte version of these characters in your singlebyte Client ID implementation. Please contact your Coremetrics Account Team or Sales Representative to discuss changes in data storage settings (singlebyte vs. multibyte).

2 Tagging Guide

2.1 *Coremetrics JavaScript Libraries*

2.1.1 Library File Placement

Coremetrics provides a single library file to support the tagging of your site: *eluminate.js*. This Coremetrics-hosted file provides JavaScript code defining a set of functions which can be called with appropriate parameter data to send data collection image requests to Coremetrics. This file must be included in all pages implementing Coremetrics.

Legacy Notes:

1. **Coremetrics began offering a single combined Coremetrics-hosted library file in April 2010 (//libs.coremetrics.com/eluminate.js). Implementations completed prior to this date may be using separate and locally hosted 'eluminate.js' and 'cmdatatagutils.js' library files.**

2. **Clients using non-hosted libraries should continue to include libraries in the page <body> section until upgrade to the Coremetrics-hosted library (version "4.7.5H" or later). Please contact Coremetrics Support to upgrade.**

- **eluminate.js**

The eluminate.js file defines the core functionality of the Coremetrics tagging technology. This file is partially minified and obfuscated. The 'tag' function definitions are not obfuscated or minified. This file should be included in the <head></head> section of all pages requiring Coremetrics tracking.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID(…);
</script>
</head>
```

- **cmcustom.js**

Your specific implementation may include customizations which override the default data collection functions or introduce new functions. This custom file, 'cmcustom.js', is delivered by Coremetrics Support and is locally hosted by your organization. Please see the contents of this file for details on your specific customizations.

This file is included in all pages immediately after the Coremetrics-hosted 'eluminate.js' library include:

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript" src="//thesite.com/cmcustom.js"></script>
<script type="text/javascript">
      cmSetClientID(…);
</script>
</head>
```

2.2 *Tagging your site*

Coremetrics provides a set of data tags that you will use to collect data from your site. These tags are defined in the Coremetrics-hosted 'eluminate.js' and/or your locally hosted cmcustom.js or cmdatatagutils.js library file.

> **Customization Note: Tag function definitions may vary by implementation. Please view the contents of your 'cmcustom.js' library file, if any, to determine what standard functions have been modified or new functions introduced to support your specific implementation. For clients hosting separate eluminate and cmdatatagutils' files, view the contents of 'cmdatatagutils' to inspect the tag definitions for your implementation.**

2.2.1 Tag Functions

To collect data the Coremetrics library files must be included in the page and calls made to the appropriate tag functions. Tags are most commonly rendered within the HTML <body> section source, but may also be called based on in-page events. Each tag has a defined list of parameters provided at the time of function calls. These functions execute to create data collection requests including both explicitly provided data parameters and automatically collected data such as timestamp, referring and destination URL's, etc. Tag parameter data is case insensitive: all collected data is converted to upper-case upon receipt and for display in processed report views.

The following 'tag' function calls are applicable to Coremetrics implementations for all business verticals:

- cmCreatePageviewTag()
- cmCreateRegistrationTag()
- cmCreateElementTag()
- cmCreateConversionEventTag()

The following 'tag' function calls are applicable to Coremetrics implementation for sites within the Retail, Travel and Financial Services business verticals:

- cmCreateProductviewTag()
- cmCreateShop5Tag()
- cmCreateShop9Tag()
- cmCreateOrderTag

2.2.2 Tag Placement

All Coremetrics tag function calls should be placed within the <body> section of the page being tagged, as close as possible to the end of the body section (the </body> tag). To achieve optimal results across all Coremetrics products and services, tag function calls made from the HTML source code of a page should be placed at the very end of the <body> section. Coremetrics tags may also be called in response to visitor interaction with in-page/site elements.

```
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("HOME PAGE","HOME");
</script>
</body>
```

2.3 *Setting Data Collection Parameters*

Coremetrics provides several functions allowing control of various aspects of the Coremetrics implementation.

- **cmSetClientID(...)** - this function is required for all implementations.

- **cmSetupOther(…)** - optional function allowing control of various optional implementation parameters. If this function is not called default settings will be used for the implementation. Coremetrics Support will advise you if this function is required for your implementation and what parameter values to pass.
- **cmSetupNormalization(…)** - optional function for implementation of any needed normalization of query string parameters from collected tag referring and destination URL values, or automatically collected 'linkclick' tag URL values. Coremetrics Support will advise if this function is required for your implementation and what parameter values to pass.

  Legacy Note: These functions may not be available or required for implementations completed prior to April 2010. Please contact Coremetrics Support to upgrade. Implementations completed prior to April 2010 may be using a slightly different cmSetClientID function which requires only 3 parameters (Client ID, Client Managed and Cookie Domain).

2.3.1 cmSetClientID(…)

This function is required and sets values for 'Client ID', 1st Party method ('Client Managed' or 'Coremetrics Managed'), 'Data Collection Domain' and 'Cookie Domain'. This function must be called on every page sending data to Coremetrics.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      // Client Managed First Party
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");

      // Coremetrics Managed First Party
      cmSetClientID("99999999",false,"<dcd>.thesite.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("FAQ Page 1", "FAQ");
</script>
</body>
```

- **Sending Data Tag Requests to Multiple Client IDs**

It is possible to copy data tags to two or more Client ID's by specifying a semicolon-delimited list of ID values. The IDs specified must all be production IDs or test IDs: a mix of test and production IDs is not allowed. If sending tags from multiple sites where each site also copies tags to a dedicated 'aggregate' ID, the first ID specified in the list should be the 'aggregate' ID. The 'aggregate' ID reports data tags copied from multiple sites.

- Example of sending tag requests from site #1 to the 'aggregate' ID and site-specific ID #1, where 99999999 is the 'aggregate' ID and 11111111 is the site-specific ID:

```
cmSetClientID("99999999;11111111",true,"data.coremetrics.com","site1.com");
```

- ... and from site #2 to the 'aggregate' ID and site-specific ID #2 (ID 22222222):

```
cmSetClientID("99999999;22222222",true,"data.coremetrics.com","site2.com");
```

Note that copying requests to two or more ID's increases total server calls assessed by (1*the number of IDs specified). In the example above, each tag sent from the page will be copied to 2 ID's, resulting in 2x total server calls assessed.

Coremetrics Account, Sales and Provisioning teams will be able to provide specific information regarding what ID's, if any, have been contracted for 'aggregate' data collection and the Monthly Million Server Call volume contracted for each ID.

Parameters

Parameter	Required	Description
Client ID	Required	Unique 8-digit Coremetrics-assigned account code associated with a single analytics data warehouse and reporting instance.
Client Managed	Required	Boolean true or false: true indicates 'Client Managed' 1st Party. false indicates 'Coremetrics Managed' 1st Party'.
Data Collection Domain	Required	The target domain for Coremetrics data collection requests. **If Client Managed = false:** Data Collection Domain should be set to the Coremetrics Managed 1st Party data collection domain. Example: ("<1stpartydcd>.thesite.com"). **If Client Managed = true:** Data Collection Domain should be set to "data.coremetrics.com".

Parameter	Required	Description
Cookie Domain	Required	The domain for Client Managed Coremetrics cookies. Cookie Domain should be set to the 2nd level site domain ("thesite.com") of the domain serving the tagged page(s). In cases of extra domain levels prior to the uniquely identifying site domain value, this may be a 3rd or higher level. Example: "thesite.co.uk". • **If Client Managed = false:** Coremetrics Visitor and Session cookies will be set under the Coremetrics Managed Data Collection Domain ("<1stpartydcd>.thesite.com"). Other Coremetrics cookies will be set in the Cookie Domain. • **If Client Managed = true:** ALL Coremetrics cookies are set in the Cookie Domain.

2.4 *Test vs. Production Environments*

Coremetrics provides two environments: test and production. The test environment should be used while tags are in development (your development and staging environments). The production environment should be used once the tags are moved to the live production site.

2.4.1 Test System Details

1. **Data Collection Domain:** testdata.coremetrics.com
2. **Client ID:** your 8-digit production Client ID, with first digit replaced by "6". Example: "12345678" production ID has a test ID equivalent of "**6**2345678"
3. **Reporting URL:** http://welcome.coremetrics.com

2.4.2 Sending Data to the Test System

In order to send data collection to the 'test' 6-series Client ID and Data Collection Domain, the cmSetClientID(…) function call on non-production pages should be modified.

- The following page is pointed to to Test reporting. Note the first digit of the Client ID parameter is changed to "6" and Client Managed parameter is always set to false when sending data to test.

```
<head>
```

```
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("69999999",false,"testdata.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("FAQ Page 1", "FAQ");
</script>
</body>
```

- The following page is pointed to Production reporting.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("FAQ Page 1", "FAQ");
</script>
</body>
```

2.5 *Data Tags*

The data collection tag function definitions documented in this section apply to standard Coremetrics implementation using the Coremetrics-hosted 'eluminate' library file. Tag function definitions may vary by implementation for clients with customizations. Please view the contents of any locally hosted 'cmdatatagutils' or 'cmcustom' library file to determine if any standard functions have been modified or new functions introduced to support your specific implementation.

The following values are automatically removed from Coremetrics tag parameter data and cannot be collected or reported:

1. single quote (')
2. double quote (")
3. carriage return (hex 0D; regular expression \r)
4. line feed (hex 0A; regular expression \n)
5. comma (,): replaced with a space character prior to data loading and reporting

2.5.1 Page View Tag

Description

The Page View tag is used to capture clickstream data as the visitor moves from page to page within the site. A Page View tag tells Coremetrics that someone has viewed a page uniquely identified by the Coremetrics 'Page ID' collected as parameter 1.

The Page View tag also captures data related to onsite keyword searches. On search results pages, the Search Term parameter of the Page View tag should be set to the value of the term on which that search was performed. The Search Results parameter should be set to the number of results returned by the search.

Key Reports Populated: Dashboards; Site Metrics; Marketing; Content (Page Categories, * Pages, On-Site Search); Paths; Demographics (Geography, Languages, Profile Segments); Report Segments

Tagging Function

In order to use the Page View tag, a call is made to cmCreatePageviewTag(…) function with the appropriate parameters.

Parameters

Parameter	Required	Description	Length
Page ID	Required	Uniquely identifies the given 'page' in Coremetrics. This can be any alphanumeric string and should be set according to the agreed upon page naming conventions.	256
Category ID	Optional	Category ID for the leaf node to which this page belongs. This should match with a category ID sent in the CDF file.	256
Search Term	Optional	Onsite search term used to generate the search results page.	256
Search Results	Optional	Number of results returned by the keyword search. This value should equal the total results for this search across all results pages. If the search returned no results, this value should be "0".	10
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per value

Extra Fields	Optional	Up to 15 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, the first 15 values can be automatically copied from the first 15 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

- **Examples – Retail and Content**

The following is an example of creating a Page View tag with a Page ID of "FAQ Page 1", no onsite search term or results, and a Category ID of "FAQ".

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
       cmCreatePageviewTag("FAQ Page 1", "FAQ");
</script>
</body>
```

The next example creates a Page View tag for a search results page where the search term "jeans" was used and 100 results were returned, where <N> is the specific results page browsed in a multi-page results set. The search results value should include total results across all pages.

```
<script type="text/javascript">
       cmCreatePageviewTag("Search Successful: Page <N>", "SEARCH", "jeans", "100");
</script>
<body>
```

The next example creates a Page View tag for a search results page where the search term "xyz" was used and "0" results were returned: the search term returned no results.

```
<script type="text/javascript">
       cmCreatePageviewTag("Search Unsuccessful", "SEARCH", "xyz", "0");
</script>
<body>
```

- **Example – Travel**

```
cmCreatePageviewTag("Hotels", "US:TX:Austin");
```

- **Example – Travel Search Results**

```
cmCreatePageviewTag("Search Successful>Hotels: Page 1", "HOTEL-SEARCH",
"location:austin", "14");
```

- **Example – Financial Services**

```
cmCreatePageviewTag("APPLICATION STEP 1(START):Home Equity Loan", "Consumer Loan");
```

- **Example – Financial Services Search Results**

```
cmCreatePageviewTag("Search Successful: Page 1", "SEARCH", "home equity", "100");
```

2.5.2 Product View Tag

Description

The Product View tag captures information about views of product detail pages. The Product View tag should be called on the lowest level detail page for products, which is typically the Product Details page.

The default Product View tag populates the 'Product Views' metric for the product specified and does not populate Content reporting. A separate 'Page View' tag should be sent from the Product Detail page (see Section 2.6). If necessary, more than 1 'Product View' tag may be sent from a single page to track views of multiple related product details which can be simultaneously selected and carted in specific quantities. For example a page displays a dining set with 2 related products that can be selected in any quantity together or seperately: 'chair 123' and 'table ABC'. Upon render of the page, a single 'Page View' tag and two 'Product View' tags will be collected: one for 'chair 123' and the second for 'table ABC'. Note this does not apply to 'category listing' type pages which display multiple results for a 'type' or 'category' of product: these pages should be tracked with a single 'Page View' tag.

> **Customization Note: Some implementations of Coremetrics may count the 'Product View' as a 'Page View' and additionally populate Content reporting. In this case no additional 'Page View' tag should be sent. If the Product View tag also counts as a 'Page View', the parameter 'Is Page View? (pc):"Y"' will appear in the displayed tag when viewed in Coremetrics Tagbar (see Section 5.1). Only one 'Product View' should be sent per page for these implementations. Contact Coremetrics Customer Support if a change is needed in the definition of your 'Product View' tag.**

Key Reports Populated: Dashboards; Site Metrics; Products; Paths; Demographics (Profile Segments); Report Segments

Tagging Function

In order to use the Product View tag, a call is made to cmCreateProductviewTag(…) function with the appropriate parameters.

Parameters

Parameter	Required	Description	Length
Product ID	Required	Product ID	256

Parameter	Required	Description	Length
Product Name	Required	Name of the product being viewed.	256
Category ID	Optional	Category ID for the leaf node to which this product belongs. This should match with a category ID sent in the CDF file.	256
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per attribute
Virtual Category (cm_vc)	Optional	This value will cause all activity related to this Product ID collected with 'Product View', 'Shop Action 5' and 'Shop Action 9' tags in this session to be categorized according to the value sent, regardless of other 'Category ID' values that may be associated with this Product ID in this session. This parameter facilitates specification of default session categorization for a Product ID based on viewing of the product detail page from a specific placement link, such as a 'you might also like', 'may we suggest' or other 'Cross-Sell' type placement. In cases where multiple values are collected for the same Product ID in a given session, the last 'Virtual Category' collected in the session will be used to categorize activity for that Product ID.	256

- **Examples – Retail and Content**

The following is an example of code to create a Product View tag for a product with a Product ID of "12345", a Product Name of "Product X", and a Category ID of "CATXYZ". Note the additional 'Page View' tag collecting 'Content' metric for the page:

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
```

```
      cmCreatePageviewTag("PRODUCT: Product X (12345)", "CATXYZ");
      cmCreateProductviewTag("12345", "Product X", "CATXYZ");
</script>
</body>
```

- **Example – Travel**

```
      cmCreatePageviewTag("TICKET: ROUTE: Los Angeles(LAX) to Bangkok(BKK)(LAX-BKK)",
"airline reservations");
      cmCreateProductviewTag("LAX-BKK", "ROUTE: Los Angeles(LAX) to Bangkok(BKK)", "airline
reservations");
```

- **Example – Financial Services**

```
cmCreatePageviewTag("PRODUCT: APPLICATION:Home Equity Loan (12345)", "consumer loans");
cmCreateProductviewTag("12345", "APPLICATION:Home Equity Loan", "consumer loans");
```

2.5.3 Technical Properties Tag

Description

The Technical Properties tag gathers technical information about the visitor's system. This includes browser type/version, operating system/version, monitor resolution and depth, and JavaScript version among other values.

Key Reports Populated: Dashboards (Design Overview); Report Segments; Demographics (Languages/Time Zones); System (Browsers/Operating Systems/Screen Resolution/Color Depth/Java Availability/JavaScript Version/Netscape Plugins)

The Technical Properties tag is automatically collected on the first page of the visitor's session. Coremetrics automatically converts the pageview tag found on the first page of the visitor session to a 'Technical Properties' tag, retaining the original Page ID and other tag parameters, but adding additional information about the visitors OS and browser. There is no need to manually implement a call to the Technical Properties tag function as this is handled automatically by Coremetrics.

- The automated technical properties feature is only available with eluminate.js version 4.1.2 or later. Prior library versions require manual execution of the Technical Properties tag and loading of a third library, techprops.js.
- A library update may be required to enable this feature if originally implemented prior to September 2007
- Please contact Coremetrics Support if unsure of which library version you are using.

2.5.4 Shop Action 5 Tag

Description

The Shop Action 5 tag captures data about selected products and which products are present in a shopping cart, if any, when the cart is viewed. Whenever the customer selects a product or views a shopping cart with carted products, a Shop Action 5 tag should be called for the selected product or each displayed carted product including information about the products such as Product ID, quantity, and price. In cases where a customer can select or cart a product without actually being directed to the cart page, a single Shop Action 5 tag should be called for the product just added. For linear purchase or application processes without a 'cart', the Shop Action 5 tag should be sent when the product is 'selected'. Typically this occurs at the earliest point in time when the selected product ID, unit price, and unit quantity values are known.

Key Reports Populated: Dashboards (Commerce Overview); Site Metrics; Products (all); Paths; Demographics (Profile Segments); Report Segments

Tagging Function

In order to use the Shop Action 5 tag, a call is made to cmCreateShopAction5Tag(…). This call should be made for the product selected or for each product viewed in the cart.

In addition to these function calls, you must also make a single call to the function cmDisplayShop5s() at the end of the sequence of 1 or more cmCreateShopAction5(…) tags. This function evaluates the Shop Action 5 tag data and performs any needed client side aggregation of identical products. The ShopAction5 data collection image request(s) will be sent only when the cmDisplayShop5s() function is called.

Parameters

Parameter	Required	Description	Length
Product ID	Required	Product ID	256
Product Name	Required	Name of the product in the cart.	256
Quantity	Required	Quantity of this product currently selected or viewed in a cart. For Travel this is typically the # of tickets selected for a given trip or event, the # of items selected for a given rental-product, or the # or room-nights selected for booking in the hotel. For Financial Services applications this is typically "1", if only 1 quantity is applicable to a given financial product.	8

Parameter	Required	Description	Length
Unit Price	Required	Price of each unit of the product. This value should be a decimal number and should not include a dollar sign ($). For Travel this is typically the price for a given route or event ticket, rental product, hotel room/night price, etc. For Financial Services applications this is usually "0" if no revenue value is be associated with the application.	16.2
Category ID	Optional	Category ID for the leaf node to which this product belongs. This should match with a category ID sent in the CDF file.	256
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. Shop Action 5 and 9 tags both share the same 50 attributes. See Appendix 8.4 for more details.	100 per value
Extra Fields	Optional	Up to 15 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, the first 15 values can be automatically copied from the first 15 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

Examples

The following example shows Shop Action 5 tags being sent for a couple of products on the Cart page:

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
       cmCreatePageviewTag("Cart", "CART");

       // A separate call to cmCreateShopAction5Tag should be made for each cart entry
       cmCreateShopAction5Tag("12345", "Product X", "2","5000.42","CATXYZ");
       cmCreateShopAction5Tag("67890", "Product Y", "1", "10.95", "CATABC");
       cmDisplayShop5s();
</script>
</body>
```

- **Examples – Travel: 2 hotels simultaneously selected for booking (1 and 2 rooms respectively)**

```
cmCreatePageviewTag("Booking: Billing Page", "Booking");

cmCreateShopAction5Tag("12345", "Property ABC", "1", "49.99", "US:TX:Austin");
cmCreateShopAction5Tag("67890", "Property XYZ", "2", "65.49", "US:TX:Austin");
cmDisplayShop5s();
```

- **Examples – Financial Services: selected application and started application process**

```
cmCreatePageviewTag("APPLICATION STEP 1(START):Home Equity Loan", "consumer loans");

cmCreateShopAction5Tag("12345", "APPLICATION:Home Equity Loan", "1", "0", "consumer
loans");
cmDisplayShop5s();
```

2.5.5 Shop Action 9 Tag

Description

The Shop Action 9 tag captures data about what products were purchased by a customer. Like the Shop Action 5 tag, one tag should be sent for each product line item purchased. These tags should be sent on the receipt or other completion page confirming a successful order. Shop Action 9 and Order tags will be rejected if the same order ID is found in the lifetime data for that Coremetrics Client ID.

Key Reports Populated: Dashboards (Commerce Overview); Site Metrics; Products; Paths; Demographics (Profile Segments); Report Segments

Tagging Function

In order to use Shop Action 9 tags, a call is made to cmCreateShopAction9Tag(…) for each product purchased. In addition, a single call to cmDisplayShop9s() must be made after all calls to cmCreateShop9Tags(…) in order to actually send the data collection image request(s).

Parameters

Parameter	Required	Description	Length
Product ID	Required	Product ID	256
Product Name	Required	Name of the product.	256

Quantity	Required	Quantity of this product purchased. For Travel this is typically the # of tickets purchased for a given trip or event, the # of items rented for a given rental-product, or the # or room-nights booked for this hotel. For Financial Services applications this is typically "1", if only 1 quantity is applicable to a given financial product.	8
Unit Price	Required	Price of each unit of the product. This value should be a decimal number and should not include a dollar sign ($). For Travel this is typically the price for a given route or event ticket, rental product, hotel room/night price, etc. For Financial Services applications this is usually "0" if no revenue value is be associated with the application upon completion.	16.2
Customer ID	Required	Customer ID for the customer who purchased the product. This should match the Customer ID field in the accompanying Order Tag.	256
Order ID	Required	Order ID for the order to which this line belongs. This should match the Order ID in the accompanying Order Tag.	64
Order Subtotal	Required	Subtotal for the order to which line item belongs. This should not include Shipping and Handling or Tax and should match the Order Subtotal in the accompanying Order Tag. This value should be a decimal number and should not include a dollar sign ($). The Order Subtotal should match the sum of Quantity * Unit Price for all ShopAction9 tags (products) purchased in this order.	16.2
Category ID	Optional	Category ID for the leaf node to which this product belongs. This should match with a category ID sent in the CDF file.	256

Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. Shop Action 5 and 9 tags both share the same 50 attributes. See Appendix 8.4 for more details.	100 per value
Extra Fields	Optional	Up to 15 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, the first 15 values can be automatically copied from the first 15 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

Examples

The following example shows the use of the Shop Action 9 tag on the Order Thank You page.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
	cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
	cmCreatePageviewTag("Order Thank You", "CART");

	// A separate call to cmCreateShopAction9Tag should be made for each cart entry
	cmCreateShopAction9Tag("12345", "Product X", "2","5000.42", "cust123", "order123",
"10011.79", "CATXYZ");
	cmCreateShopAction9Tag("67890", "Product Y", "1", "10.95", "cust123", "order123",
"10011.79", "CATABC");
	cmDisplayShop9s();
	cmCreateOrderTag("order123", "10011.79", "5.95", "cust123", "Austin", "TX", "78727");
	cmCreateRegistrationTag("cust123", "customer@mail.com", "Austin", "TX", "78727",
"US");
</script>
</body>
```

- **Examples – Travel: 2 hotels simultaneously booked (1 and 2 rooms respectively)**

```
	cmCreatePageviewTag("Booking: Thank You", "Booking");

	cmCreateShopAction9Tag("12345", "Property ABC", "1", "49.99", "cust123", "booking123",
"180.97", "US:TX:Austin");
	cmCreateShopAction9Tag("67890", "Property XYZ", "2", "65.49", "cust123", "booking123",
"180.97", "US:TX:Austin");
	cmDisplayShop9s();
```

- **Examples – Financial Services: application completed**

```
	cmCreatePageviewTag("APPLICATION STEP 5(COMPLETION):Home Equity Loan", "consumer
loans");
```

```
      cmCreateShopAction9Tag("12345", "APPLICATION:Home Equity Loan", "1", "0", "cust123",
"application123", "0", "consumer loans");
      cmDisplayShop9s();
```

2.5.6 Order Tag

Description

The Order tag captures order header information such as customer ID, order ID, order subtotal, and shipping and handling. The Order tag should be sent on the receipt page confirming order completion. Shop Action 9 and Order tags will be rejected if the same order ID is found in the lifetime data for that Coremetrics Client ID.

Key Reports Populated: All aggregate 'Sales' and 'Order' metrics outside of the Products Reporting; Demographics (Profile Segments)

Tagging Functions

In order to use the Order tag, a call is made to cmCreateOrderTag(…) function. This call should be made *after* all calls to cmCreateShop9Tag(…) and cmDisplayShop9s().

Parameters

Parameter	Required	Description	Length
Order ID	Required	Order ID for this order. This should match the Order ID sent in the Shop 9 tags for the line items in the order.	64
Order Subtotal	Required	Subtotal for this order. This should not include shipping and handling or tax and should match the Order Subtotal sent with the related Shop 9 tags for this order. This value should be a decimal number and should not include a dollar sign ($). The Order Subtotal should match the sum of Quantity * Unit Price for all ShopAction9 tags (products) purchased in this order.	16.2
Order Shipping	Required	Shipping and Handling for this order. For Travel or Financial Services, this will typically be "0", but may be used to track other relevant monetary data such as fees. Coremetrics reporting will reference this as 'Shipping'.	16.2

Parameter	Required	Description	Length
Customer ID	Required	Customer ID for the customer who placed the order. This should match the Customer ID sent in the Shop 9 tags for the line items in the order, as well as the Customer ID sent in the Registration Tag.	256
Customer City	Optional	City of the Billing Address for this customer.	256
Customer State	Optional	State of the Billing Address for this customer.	256
Customer Postal Code	Optional	Postal Code of the Billing Address for this customer.	256
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per value
Extra Fields	Optional	Up to 15 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, the first 15 values can be automatically copied from the first 15 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

Examples

The following is an example of the Order tag being called on the Order Thank You Page:

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
       cmCreatePageviewTag("Order Thank You", "CART", null, null);
       // A separate call to cmCreateShopAction9Tag should be made for each cart entry
       cmCreateShopAction9Tag("12345", "Product X", "2","5000.42", "cust123", "order123",
"10011.79", "CATXYZ");
       cmCreateShopAction9Tag("67890", "Product Y", "1", "10.95", "cust123", "order123",
"10011.79", "CATABC");
       cmDisplayShop9s();

       cmCreateOrderTag("order123", "10011.79", "5.95", "cust123", "Austin", "TX", 78727");
       cmCreateRegistrationTag("cust123", "customer@mail.com", "Austin", "TX", "78727",
"US");
</script>
</body>
```

- **Examples – Travel: 3 hotel rooms booked (2 hotels)**

```
	cmCreatePageviewTag("Booking: Thank You", "Booking");

	cmCreateShopAction9Tag("12345", "Property ABC", "1", "49.99", "cust123", "booking123",
"180.97", "Region:US:TX:Austin");
	cmCreateShopAction9Tag("67890", "Property XYZ", "2", "65.49", "cust123", "booking123",
"180.97", "Region:US:TX:Austin");
	cmDisplayShop9s();

	cmCreateOrderTag("booking123", "180.97", "0", "cust123", "Austin", "TX", 78727");

	cmCreateRegistrationTag("cust123", "customer@mail.com", "Austin", "TX", "78727",
"US");
```

- **Examples – Financial Services: application completed**

```
	cmCreatePageviewTag("APPLICATION STEP 5(COMPLETION):Home Equity Loan", "consumer
loans");

	cmCreateShopAction9Tag("12345", "APPLICATION:Home Equity Loan", "1", "0", "cust123",
"application123", "0", "consumer loans");
	cmDisplayShop9s();

	cmCreateOrderTag("application123", "0", "0", "cust123", "Austin", "TX", 78727");

	cmCreateRegistrationTag("cust123", "customer@mail.com", "Austin", "TX", "78727",
"US");
```

2.5.7 Registration Tag

Description

The Registration tag creates a Lifetime Visitor Experience Profile (LIVE Profile) by associating a single common 'customer ID' with the Coremetrics permanent cookie set in every browser visiting the tagged site. The Registration tag also provides for collection of additional demographic information including email address, city, state/province, postal code, country and up to fifty 100 - byte custom demographic parameters. Collected data can be used to generate segmentation reporting.

Up to 50 distinct demographic values can be collected using the 'attribute' parameter of the registration tag. All 50 attributes are available in the Explore/Registration Module (if contracted). Attributes 1-15 are available for selection as Visitor Criteria in Analytics 'Manage/Report Options/Report Segments', and for LIVEmail segmentation (if contracted). Before including the collected 'attributes' in a Report Segment, the attributes must first be aliased to descriptive names within the 'Admin/Extra Fields' panel.

Attributes 11-15 are additionally available for selection as 'profile' criteria within Analytics 'Reports/Demographics/Profile Segments' reporting. Aliasing of these attributes to make them available for selection in the 'Enter Criteria' step must be requested through Coremetrics support. See Appendix 8.4 for examples of 'attribute' parameter data collection.

The Registration tag should be sent upon any of the following events, or any other event where the visitor supplies the identified 'customer ID' and any additional or updated demographics identified for collection.

- Order completion
- New account setup
- Account profile update
- Account login
- Submission of a newsletter subscription or other identifying form

Key Reports Populated: Site Metrics; Dashboards (KPI); Paths (Visitor Purchase Funnel / Previous Unique Buyer Funnel); Demographics (Profile Segments); Report Segments (Visitor Criteria); Explore/Registration Module (if contracted); LIVEmail (if contracted)

cm_lm= Registration Parameter: It is possible to generate a Registration tag automatically on marketing landing pages by specifying a 'cm_lm=<value>' query string parameter value in the marketing URL. The <value> specified in the parameter will typically be the visitor email address, obtained in conjunction with an email marketing campaign. The typical use case for cm_lm= is in marketing links placed in a marketing email. Use of cm_lm= accelerates collection of Registrant profile information in the Coremetrics data warehouse. This is particularly beneficial when used in conjunction with a LIVEmail deployment to accelerate the number of registration profiles available for generation of segmented LIVEmail feeds to ESPs.

Tagging Function

In order to call a Registration tag, a call is made to cmCreateRegistrationTag(…) function with the appropriate parameters.

> **Legacy Note: Clients implemented prior to May 2010 may be using a registration tag including custom parameters, or missing some of the parameters in the current //libs.coremetrics.com/eluminate.js function definition documented below. Please see your local 'cmdatatagutils.js' or 'cmcustom.js' file for the definition of your cmCreateRegistrationTag function.**

Parameters

Parameter	Required	Description	Length
Customer ID	Required	Customer ID for this customer.	256
Customer Email	Optional	Email address for the customer.	256
Customer City	Optional	Customer's City.	256

Customer State	Optional	Customer's State.	256
Customer Postal Code	Optional	Customer's Postal Code.	256
Customer Country	Optional	Customer's Country	256
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values representing demographic group information in Analytics and Explore Reporting. See Appendix 8.4 for more details.	100 per value

Examples

The following is an example of the Registration tag being called on the "Account Created" page.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
       cmCreatePageviewTag("Account Created", "CART");
       cmCreateRegistrationTag("cust123", "customer@mail.com", "Austin", "TX", "78727",
"US");
</script>
</body>
```

- **Examples – Retail: custom demographics through the 'attribute' parameter string: a 'membership' true/false boolean and 'indoor/outdoor preference' value.**

```
       cmCreatePageviewTag("Account Created", "CART");

       cmCreateRegistrationTag("cust123","customer@mail.com","Austin","TX","78727","US","TRUE
-_-OUTDOOR");
```

- **Examples – Travel: hotel attributes collected at time of booking: "hotel star rating-_-smoking/non-smoking-_-room type"**

```
       cmCreatePageviewTag("Booking: Thank You", "Booking");
       cmCreateShopAction9Tag("12345", "Property ABC", "1", "49.99", "cust123", "booking123",
"180.97", "Region:US:TX:Austin");
       cmCreateShopAction9Tag("67890", "Property XYZ", "2", "65.49", "cust123", "booking123",
"180.97", "Region:US:TX:Austin");
       cmDisplayShop9s();
       cmCreateOrderTag("booking123", "180.97", "0", "cust123", "Austin", "TX", 78727");

       cmCreateRegistrationTag("cust123","customer@mail.com","Austin","TX","78727","US","3-_-
SMOKING-_-DOUBLE");
```

- **Examples – Financial Services: application completed: "app-version-_-logged-in-_-pre-approval"**

```
      cmCreatePageviewTag("APPLICATION STEP 5(COMPLETION):Home Equity Loan", "consumer
loans");
      cmCreateShopAction9Tag("12345", "APPLICATION:Home Equity Loan", "1", "0", "cust123",
"application123", "0", "consumer loans");
      cmDisplayShop9s();
      cmCreateOrderTag("application123", "0", "0", "cust123", "Austin", "TX", 78727");
```

```
      cmCreateRegistrationTag("cust123","customer@mail.com","Austin","TX","78727","US","VER_
   2B-_-AUTHENTICATED-_-YES"
```

2.5.8 Error Tag

Description

The Error Tag was deprecated in January 2010. Please use the Page View tag with descriptive Page ID and Category ID parameters to track server or application error pages. Explore attributes can be used to collect additional details such as internal error codes, error message, etc.

2.5.9 Element Tag

Description

The Element tag is used to track intra-page content in Coremetrics Online Analytics. Data collected by the Element tag is used to populate values in the Element Categories and Top Viewed Elements reports. The Element tag and its associated reports provide organizations with the flexibility to track interaction with various intra-page elements and how these elements drive objective attainment. Below are some examples of elements that could be tracked using the Element tag:

Examples of Elements to Track:

Portlets o Search portlets o News portlets **Video Plays** o Play o Stop o Rewind o Explore users: see section 8.4.2	**AJAX Detail Hovers** o Product detail hover-overs o Customer review hover-overs **Dynamic Page Content Filters** o Price slider bars o Brand filter checkboxes o Feature selectors o Page or Form Error messages

Key Report Populated: Content (Elements); Report Segments

Tagging Function

In order to use the Element tag, a call is made to cmCreateElementTag(…) function with the appropriate parameters.

Legacy Note: cmCreatePageElementTag and cmCreateProductElementTag were deprecated January 2010. Calls to these functions now call cmCreateElementTag(…).

New implementations should call the 'cmCreateElementTag(…) directly. Please contact Coremetrics Support if in doubt about which function is applicable to your implementation.

Parameters

Parameter	Required	Description	Length
Element ID	Required	The unique identifier or name for the Element and the value that will be displayed in the Elements report. Length limit is in 'characters', which will vary in byte length depending on encoding.	50
Element Category	Optional	The category passed in the Element tag is used to populate the Element Categories report. Only one hierarchical level of categorization is currently supported and not related in any way to the client's Category Definition File (CDF) specification.	50
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per value

Examples

The following is an example of creating an Element tag with an Element ID of "Vacation Planner" with an Element Category ID of "Vacation Tools".

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
       cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">
       cmCreateElementTag("Vacation Planner", "Vacation Tools");
</script>
</body>
```

2.5.10 Conversion Event Tag

Description

The Conversion Event tag is employed for tracking of general non-commerce conversion events. The Conversion Event tag is used to populate values in the Conversion Events Reports and to create Key Segments. This tag and the reports it populates enable analysis of a wide variety of site activities. Below are some examples of events that could be tracked via the Conversion Event Tag:

Examples of Conversion Events:

• Increase Site Stickiness
  ○ Play online game
  ○ View account info
  ○ Use online calculator
  ○ Use trip planner
  ○ Use comparison tool
• Improve retention marketing
  ○ Register for specific newsletter
  ○ Sign up for Webinar
  ○ Add items to wish list
  ○ Set email alerts
	• Improve Self Service
	  ○ Sign up for bridal registry
	  ○ Download help documents
	  ○ Download form
	  ○ Download marketing info
	• Multichannel behavior
	  ○ Use store locator
	  ○ Visit contact us page
	  ○ Initiate chat session
	  ○ Register for callback

Key Report Populated: Events; Report Segments; Optional metric columns in Content (Page Categories and * Pages reporting, Site Promotions, Real Estate, Elements)

Tagging Function

In order to use the Conversion Event tag, a call is made to cmCreateConversionEventTag (…) function with appropriate parameters.

Parameters

Parameter	Required	Description	Length
Event ID	Required	A unique identifier for the type of conversion, such as "Account Creation" or "Special Registration". The value that is passed in the tag is the value that will be displayed in the Coremetrics reports.	256
Action Type	Required	A value of "1" or "2" depending upon whether a conversion initiation or a successful conversion completion is generated. A value of 1 should be used when an event is initiated. A value of 2 should be used when an event is successfully completed. Single-Step conversions should be represented by a value of "2".	1
Event Category ID	Optional	Allows grouping of event IDs into categories. The value that is passed in the tag is the value that will be displayed in Coremetrics' reports. The Event Category ID is self-contained and not related to the Category Definition File contents (CDF).	256
Points	Optional	A point value used in establishing an arbitrary	16

Parameter	Required	Description	Length
		"value" for a conversion. The point value allows relative weighting of and Event's 'initiation' and 'completion'. For example, a visitor initiating a low value event might be worth 5 points, whereas a visitor completing a high value event might be worth 50 points.	
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per value
Extra Fields	Optional	Up to 5 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, these values can be automatically copied from the first 5 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

Examples

- The following example shows the Conversion Event Tag used to track a Support Information Email Registration page with separate distinct start and end points.

  Support Information Email Registration

  - **Page 1** – Conversion Event Tag with Event ID = "Request More Information", Action Type = "1", Event Category ID = "Leadgen Forms", Points = "10"
  - **Page 2** – no Conversion Event Tag
  - **Page 3** – Conversion Event Tag with Event ID = "Request More Information", Action Type = "2", Event Category ID = "Leadgen Forms", Points = "20"

Code (Pages 1-3):

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</head>
<body>
...
<script type="text/javascript">

--- PAGE 1 ----
...
      cmCreateConversionEventTag ("Request More Information","1","Leadgen Forms","10");
...
--- PAGE 2 ----

      (No conversion event tag sent – call a single Pageview tag)
```

```
--- PAGE 3 ----
...
      cmCreateConversionEventTag ("Request More Information","2","Leadgen Forms","20");
...
</script>
</body>
```

- We recommend against adding Conversion Event tags on load of high volume pages as this can inflate Server Call charges. Consult your Coremetrics Account Team or Customer Support for more information about 'Server Calls'.

- The following example demonstrates tracking of a single step conversion completed from the site home page. The conversion tag is collected with Action Type "2" upon successful signup: no Action Type "1" conversion tag is collected. This results in normal reporting, but without 'abandonment' calculation for this Event. No 'Points' are collected in this example.

```
--- PAGE 1 ----  HOME PAGE
...
      (standard pageview tag sent when page loads)
...
--- PAGE 1 ----  NEWSLETTER SIGNUP COMPLETED – MESSAGE DISPLAYED
...
      cmCreateConversionEventTag ("Newsletter Signup","2","Newsletter");
...
```

2.5.11 Manual Page View Tag

Description

The Manual Page View tag is used when manual control of collected destination and destination URL values is needed. Referring and destination URL are normally collected automatically when the standard pageview tag is called (from document.referrer and window.location.href respectively).

Common Use Cases:

- Tracking of an offsite linkclick is desired in the reporting normally populated by the Page View tag. In this case the target URL of the tracked click will be collected as the 'Destination URL', while the tag 'Referring URL' will be collected as the URL of the page hosting the link (typically, window.location.pathname). This will cause the offsite content to be reported as a distinct 'Page ID' within Content and other reporting of Page View data.

- The collected referring URL must be manually assigned. This is needed in cases where document.referrer is deleted or otherwise unavailable on the first page of the session and thus cannot be collected automatically. This is important for Coremetrics attribution of the session to the correct marketing channel (Marketing Programs, Natural Search, Referring Sites). If no document.referrer value is available on first page of the session, the session will be credited to the Coremetrics Marketing "Direct Load" channel.

Key Reports Populated: Dashboards; Site Metrics; Marketing; Content (Page Categories, * Pages, On-Site Search); Paths; Demographics (Geography, Languages, Profile Segments); Report Segments

Additional Documentation: 'Coremetrics Manual Tagging for HTML.pdf' available at https://support.coremetrics.com.

Tagging Function

In order to use the Page View tag, a call is made to cmCreateManualPageviewTag (…) function with the appropriate parameters.

Parameters

Parameter	Required	Description	Length
Page ID	Required	Uniquely identifies the given 'page' in Coremetrics. This can be any alphanumeric string and should be set according to the agreed upon page naming conventions.	256
Category ID	Optional	Category ID for the leaf node to which this page belongs. This should match with a category ID sent in the CDF file.	256
Destination URL	Required	The desired destination URL value.	1024
Referring URL	Optional	The desired referring URL value. If this is not provided and the pageview is the first in the session, the session will be attributed to Coremetrics Marketing "Direct Load" channel.	1024
Attribute String	Optional	Up to 50 "-_-" delimited 'attribute' values accessible in Explore reporting. See Appendix 8.4 for more details.	100 per value
Search Term	Optional	Onsite search term used to get to the Search Results page. This should only be populated on the first Search Results page.	256
Search Results	Optional	Number of results returned by the keyword search. This is typically "1". Only sent if an onsite keyword search redirects the visitor to a standard product detail page.	10

Extra Fields	Optional	Up to 15 "-_-" delimited 'extrafield' values accessible through optional Standard Data Export. If desired, the first 15 values can be automatically copied from the first 15 Attribute String values by calling cmSetupOther ({"cmAutoCopyAttributesToExtraFields":true}); on the page prior to tag function calls.	100 per value

2.5.12 Manual Link Click Tag

Description

The Manual Link Click tag is used when the automated collection of link click data does not occur and LIVEview or Site Promotions / Real Estate reporting is desired for the affected links. Coremetrics Testing Tools including Tag Bar / Tag Monitor and Implementation Test Tool (ITT) can assist in determining if automatic Link Clicks tags are generated for the links in question. See section 5 for more information about Coremetrics Testing Tools.

Common Examples

- HTML Anchor tags without HREF= attributes or which otherwise use JavaScript to create navigation at time of click.
- Clicks on Flash, Silverlight or other interactive application elements without HTML anchors.

Key Reports Populated: LIVEview, Content (Page Categories: Context Menu / Summary Zoom / download icon: 'Download All LIVEview Data)

Additional Documentation: 'Coremetrics Manual Tagging for HTML.pdf' available at https://support.coremetrics.com.

Tagging Function

In order to use the Manual Link Click tag, a call is made to cmCreateManualLinkClickTag(…) function with the appropriate parameters.

Parameters

Parameter	Required	Description	Length
Href	Required	Value that uniquely identifies the object clicked. For standard anchors this is the link HREF= attribute value. For other objects, a different value may be used.	256

Parameter	Required	Description	Length
		If the link click is implemented for purposes of Site Promotion or Real Estate reporting, a valid relative or absolute URL must be specified including cm_sp= and/or cm_re= link parameter values.	
Link Name	Optional	Determines uniqueness of the link/object in combination with Href and Page ID. This optional value also appears in column 2 of the LIVEview Content download.	256
Page ID	Optional	Only needed if it is necessary to force association of the 'click' Href with a correct Page ID value in cases where multiple, or no, Page View tags have been sent from the page prior to the click event.	256

2.5.13 Manual Impression Tag

Description

The Manual Impression Tag is used to collect marketing 'impressions' for individual tracked Onsite Marketing Links (Site Promotions or Real Estate) in cases where automatic impression tracking cannot function or is intentionally disabled (see section 4.2). Manual Impressions are typically implemented in conjunction with a related 'Manual Link Click' tag. One Impression tag is called for each related 'Site Promotion' or 'Real Estate' link object on a given page.

Common Use Case:

- Manual 'Link Click' tags have been implemented to track JavaScript, Ajax, Flash, Silverlight or other non-HTML/HREF link objects with Site Promotions and/or Real Estate parameters: equivalent 'Manual Impression' tags must be implemented if impression tracking is desired for the related link objects.

Key Reports Populated: Marketing (Site Promotions optional 'Impressions' column); Content (Real Estate or Site Promotions optional 'Impressions' column).

Additional Documentation: 'Coremetrics Manual Tagging for HTML.pdf' available at https://support.coremetrics.com.

Tagging Function

In order to use the Manual Impression tag, a call is made to cmCreateManualImpressionTag (…) function with the appropriate parameters. An individual 'impression' is either Site Promotion or Real Estate, never both.

Parameters

Parameter	Required	Description	Length
Page ID	Required	The unique Page ID value associated with this impression. This value should match the value sent with the Page View for this page/object. This should match the Page ID of the related 'Link Click' tag.	256
Track Site Promotion	Optional	Required to track a Site Promotion Impression. This is a valid cm_sp= value: "group-_-promotion-_-link".	256
Track Real Estate	Optional	Required to track a Real Estate Impression. This is a valid cm_re= value: "version-_-area-_-link".	256

2.5.14 Form Action Tag

Description

The Form Action tag gathers information about visitor interaction with html form elements. Data collected includes form name or id, field name or id, submits, unloads and resets. Actual visitor-provided data values are not collected. The Form Action tag is generated automatically for html form objects in pages including the Coremetrics libraries and tags. There is no 'manual' equivalent of the Form Action tag.

Key Reports Populated: Content (Forms)

Metrics for each tracked form and form element will appear in reporting according to the NAME= attribute value of each tracked form. If no NAME= attribute is defined, Coremetrics will use an available ID= attribute if defined. If neither attribute is defined, the form or form element will be

reported as Form Name “UNDEFINED”. It is possible to see ‘undefined’ forms reported in combination with defined related form elements, and vice-versa.

It is possible to automatically prefix reported form names (sourced from name= or id= attribute) with the ‘Page ID’ value set on the hosting page by the Coremetrics ‘Page View’ tag. This can assist in identifying which pages forms are submitted from in cases where the same form exists across multiple ‘pages’.

- Enabling Form Action ‘Page ID’ prefixing on a page (cmSetupOther):

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("69999999",false,"testdata.coremetrics.com","thesite.com");
      cmSetupOther({"cm_FormPageID":true});
</script>
</head>
```

- Example: A form previously reported as “paymentinfo” would be reported as two seperate forms after enabling Page ID prefixing in two distinct pages: “CHECKOUT:GUEST:BILLING” and “CHECKOUT:REGISTERED:BILLING”.

1. “CHECKOUT:GUEST:BILLING_paymentinfo”
2. “CHECKOUT:REGISTERED:BILLING_paymentinfo”

2.6 *Tagging Conventions*

This section outlines a set of recommended conventions to insure that data collection provides the best possible reporting value.

2.6.1 Page ID Conventions

In order to ensure consistent, readable, and maintainable page-related reports and tagging, you will need to decide upon a set of conventions related to the naming of pages within the site.

Coremetrics uniquely identifies pages within your site based on their Page ID. This Page ID is passed to Coremetrics as a parameter in the Page View and possibly also generated automatically by the Product View tag, depending on your implementation. Pages that share the same Page ID will be seen as the same page in Coremetrics reporting.

A Page ID can be any alphanumeric string that can be built and passed into the corresponding tag function. To avoid having to statically assign a Page ID to every page in the site, you should develop page naming conventions that will allow you to generate Page IDs for every page based on a set of rules and available information.

The basic page naming convention that Coremetrics recommends clients use when applicable bases the Page ID off of the page URL. Rather than using the entire URL, which could be long and

difficult to read, the path and filename should be used. For example, this page naming convention would map the following Page ID from the given URL:

Page URL: http://www.client.com/x/y/z/thepage.html?param=1
Page ID: X/Y/Z/THEPAGE.HTML

However, this URL-based page naming convention does not work well in all situations. For example, the URL-based naming convention typically does not generate desirable Page IDs for dynamically generated template pages. If you have a template "products.asp" that is used to display all product details pages, rather than naming the page "products.asp", you should instead name it based off the specific product being displayed to the user. In this specific example, Coremetrics suggests a convention of "PRODUCT: <product_name> (<product_id>)", where <product_name> and <product_id> would be replaced with the appropriate values.

Prior to the start of tagging, you should examine your site to determine which types of pages will need naming conventions other than the URL-based convention. For those pages, you should lay out rules similar to the product page rule described above.

Page Type	Convention
Product View Pages	"Product: <product_name> (<product_id>)"
Category Pages	"Cat: <category_name>"
Successful Search Results Page	"Search Results: Successful page <page_num>"
Unsuccessful Search Results Page	"Search Results: Unsuccessful"

2.6.2 Product ID Conventions

Coremetrics captures Product level data through the Product View, Shop Action 5, and Shop Action 9 tags. This data allows you to see when items are viewed, added to cart/selected and purchased. Because Coremetrics allows flexibility in determining what constitutes a product in the Coremetrics reporting system, you must determine at what level of detail products will be tracked.

Coremetrics uniquely identifies a product based on the value passed as the Product ID. The Product ID can be any alphanumeric string that uniquely identifies a product. Before doing any product related tagging, you must determine what should be considered a "product". Should a product be a particular SKU, or should it be a style or family of SKUs? While using the SKU as the Product ID may seem like the most obvious choice for the Product ID, often using a family or style instead is preferable. For example, a clothing retailer with a SKU for every size and color of a particular shirt may not want to use the SKU as their Product ID because they will end up with unmanageable numbers of products in their reporting. Rather than report at such a granular level, the retailer would instead want to track at the shirt's style level, so that all sizes and colors of a

particular style of shirt would be tracked as the same item. In this case, the retailer might choose to use the Style ID as the Product ID value instead of the SKU.

The best gauge for what level the product should be tracked is to look at which level the Product Details page exists. In cases where the Product Details page lives at the family or style level, that ID should be used. If the Product Details page lives at the SKU level, meaning there is a separate page for each SKU, the SKU probably makes the most sense as the Product ID.

The value that you decide upon for your Product ID must be available whenever the Product View, Shop Action 5, and Shop Action 9 tags are called. The same Product ID should be used in all tags for a given product.

2.6.3 Registration Conventions

The Registration tag allows you to associate a visitor's demographic information with their Coremetrics cookie ID. Coremetrics identifies a visitor using the Customer ID provided in the Registration tag. The Customer ID can be used to link activity from one visitor across multiple computers. Before tagging the site with the Registration tag, you must determine what value to use as the Customer ID.

The Customer ID can be any alphanumeric string that is relatively long-lived and consistent for a given user from visit to visit. It must be a value that is available whenever the Registration tag needs to be called (most likely when an order is completed, a new account is set up, or existing account information is updated). Possible options include an internal customer ID or the visitor's email address.

2.7 *Server-side include files and flags*

As a best practice, all Coremetrics-related code should be modularized so that it is easy to maintain and deactivate if necessary. In order to do this, Coremetrics recommends creating a single Coremetrics server-side include file that can be used on all dynamic pages within your site. This include file will contain logic to determine what type of page is being rendered, and based on that type, write the appropriate Coremetrics JavaScript code into the page. The include file should also have a flag that will allow you to switch off the Coremetrics JavaScript code from rendering, in case you ever need to turn off the Coremetrics code for any reason, as well as an automated way of determining whether to point the tags to the Coremetrics test or production servers.

The Coremetrics include file should be included in a global header, global footer, or other global include file used in the site. This will enable the code to be immediately propagated to all pages that make use of this include. Having access to these global includes will allow you to avoid having to touch a large number of pages in the implementation process.

2.7.1 Main Case Statement

A Case statement may be used in your logic to determine what type of page is being rendered and write the appropriate Coremetrics JavaScript code. The case statement should have a number of checks for each type of page that will need tags other than the default Page View tag. Each of these checks would then render in the appropriate tagging functions needed on that page type. The

default case for pages that don't fall into any special cases would be to throw a Page View tag with the default naming convention.

The following is an example of a case statement in pseudocode:

```
if (pageType is product details page) {
    render cmCreateProductViewTag() with appropriate parameters
} else if (pageType is shopping cart page) {
    render cmCreatePageviewTag(), cmCreateShop5Tag(), and cmDisplayShop5s() with appropriate
parameters
} else if (pageType is order confirmation page) {
    render cmCreatPageviewTag(), cmCreateShop9Tag(), cmCreateOrderTag(), cmDisplayShop9s(),
cmCreateRegistrationTag() with appropriate parameters
} else ...
    ...Do other page type checks here...
} else {
    default case, render cmCreatePageviewTag() with default naming convention
}
```

2.7.2 On/Off Flag

In case you ever need to turn off the Coremetrics tracking on your site, you should implement an on/off server-side flag that will allow you to specify whether or not to render the Coremetrics code into the site. All Coremetrics code should be wrapped in a case statement that checks if this flag is true before executing.

Expanding on the case statement example, the following shows the implementation of an on/off flag in pseudocode:

```
if(coremetricsOnFlag) {

  render cmSetClientID(…) call

  if (pageType is product details page) {
    render cmCreateProductViewTag() with appropriate parameters
  } else if (pageType is shopping cart page) {
    render cmCreatePageviewTag(), cmCreateShop5Tag(), and cmDisplayShop5s() with appropriate
parameters
  } else if (pageType is order confirmation page) {
    render cmCreatPageviewTag(), cmCreateShop9Tag(), cmCreateOrderTag(), cmDisplayShop9s(),
       cmCreateRegistrationTag() with appropriate parameters
  } else ...
    ...Do other page type checks here...
  } else {
    default case, render cmCreatePageviewTag() with default naming convention
  }
}
```

2.7.3 Test/Production Flag

If you are using the same code base in both your development and production environments, it is best practice to implement a condition that determines which server (development, staging, or production) the code is rendering and call the cmSetClientID(…); function with appropriate values (clients not using the Coremetrics-hosted library must also call cmSetProduction();). For example, create a server-side flag to indicate if the code is on the development, staging, or production server.

If you can not create the server-side flag, a client-side flag can be implemented in JavaScript to call cmSetClientID(…); based on URL domain value for example. However, using the URL is a much less dependable method of determining test vs. production. You must maintain the URL-based rules to encompass all possible URLs in the Production environment.

Example:

```
if(coremetricsOnFlag) {
  if(productionServerFlag)  {
     cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
  } else {
     cmSetClientID("69999999",false,"testdata.coremetrics.com","thesite.com");
  }
…
```

2.8 *Tagging Framed Pages*

Framed pages may require the inclusion of an additional Coremetrics library file within the parent frameset. The file 'cmframeset.js' contains code that will properly set the referring URL and pass on important query string parameters from the parent frameset to the first Page View of its children. Please contact Coremetrics Support to obtain this file.

2.8.1 Referral URL

The Coremetrics tagging uses the referral URL that the browser provides. With framed pages, browsers consider the parent frameset to be the referral URL for all child pages within a frameset rather than the last page viewed. The cmframeset.js file contains code to properly set the referral URL within framed pages to be the URL of the last page viewed.

2.8.2 Query String Parameters

Certain query string parameters for parent framesets need to be captured within Coremetrics Page View tags. These query string parameters never get passed on to Coremetrics because the parent frameset is not tagged. In order to capture these parameters, they must be passed into the first Page View tag sent by any of its children.

Marketing Management Center (MMC) cm_mmc= parameters are a good example. External campaigns that point to a framed page will need to contain cm_mmc= parameters within their query string in order to attribute clickthroughs and activity to the campaign. These parameters must be captured by the first Page View in the visitor's session. Because the parent frameset contains the MMC parameters in its destination URL but does not send a tag, the MMC parameters must be passed on to one of the child frames that do throw a tag. Coremetrics can provide a 'cmframeset.js' file containing code to parse out the MMC parameters and attach them to the URL of the first Page View tag called by its children. This enables correct MMC tracking. Please contact Coremetrics Support to request this file.

Other query string parameters can be passed on to the children Page Views in the same manner. A simple change must be made to the cmframeset.js file. Please consult Coremetrics Support for more information on making this change if necessary.

2.8.3 Third Party Frames

Child frames served by 3rd party domains (domain does not match the domain serving the parent frame) should include the Coremetrics src=libraries again within the 3rd party frame. This prevents cross-domain scripting errors by insuring tag functions are defined within the 3rd party domain scope. Note that the cmSetClientID(…) function call in the new 3rd party frame should reference the 3rd party domain in parameter 4 (Cookie Domain).

3rd party frames cannot access the parent frame due to cross-domain scripting limitations imposed by browsers. Therefore it will not be possible to use cmframeset.js to transfer parent frame referring URL query string parameters to a pageview tag in the 3rd party child frame. In this case a Page View tag should be sent from the parent frame to insure marketing program information can be captured.

2.8.4 Examples

The following is an example of a frameset that includes the cmframeset.js file.

```
<html>
<script type="text/javascript" src="cmframeset.js"></script>
<frameset cols="50%,50%">
       <frame src="body.html" name="body">
       <frameset rows="100,*">
              <frame src="nav.html" name="nav">

<frame src="footer.html" name="footer">
       </frameset>
</frameset>
</html>
```

2.9 *Server Calls*

Coremetrics tracks data collection usage through 'Server Calls' assessed according to the type and number of data tags collected in the reporting period. Server calls are tracked in the Coremetrics Analytics interface Top LineSummary / 'Server Calls' report.

Tag Type	Server Calls
Page View, Product View, Shop (5 or 9), Order, Element, Technical Properties, Error	1 Note: a Product View tag modified to count as a pageview (pc="Y") will be assessed 2 total server calls. (tag populates both the Page and Product Categories reports).

Tag Type	Server Calls
Impression	1/10th to 2/5th (.1 to .4 server call) Per Site Promotion, Real Estate or WebSphere Commerce E-Marketing Spot Impression collected: 10 impressions = 1 to 4 server calls. The exact server calls charged varies based on the total combinations of unique SP or RE parameter values processed. If Site Promotion and Real Estate are tracked for the same link, this counts as 2 impressions.
Registration	0
Link Click, Form Action	0 These are tags generated automatically in response to visitor clicks on anchor tags, or submission of forms, in pages including the Coremetrics libraries and at least 1 Page View tag (or equivalent).
Conversion	1 Each unique 'conversion' sequence within a single session. A single conversion 'sequence' is one or two related Conversion tags received in the session: Case 1: a single Conversion tag with Action Type "1" for a given Event ID and Event Category ID combination. Case 2: a single Conversion tag with Action Type "2" for a given Event ID and Event Category ID combination. Case 3: two Conversion tags in the session with the same Event ID and Event Category ID, one having Action Type "1" and the other Action Type "2". In other words, an initiation and completion for the same event is assessed only 1 total server call.

Tag Type	Server Calls
Custom Tag	1 These are tags populating custom reports or custom exports exclusively (tid=7). In addition, any standard tag also collecting "ps" custom data parameters incurs an additional server call charge (1). Example: a standard 'Page View' tag collecting 1 or more "ps" custom parameters would incur 2 total server calls.

3 Categorization

Coremetrics allows you group your site content and/or products into categories for reporting. These categories are managed through a combination of the Category IDs in the tag functions and the Category Definition File, an offline file upload. There are two types of categorization within Coremetrics: Product categorization and Page categorization. Product categorization is captured in the Product View, Shop Action 5, and Shop Action 9 tags and displayed within Product Categories. Page Category data is captured in the Page View and Technical Properties tags and displayed within the Page Categories report.

3.1 *Category IDs*

Category IDs are captured in the Page View, Product View, Shop Action 5, Shop Action 9, Technical Properties, and Error tags. The Category ID assigns that particular page or product related action to a particular category. The Category ID only specifies the immediate leaf category that the action belongs to and does not contain within it any sense of a larger hierarchy. The Category ID for a given product should be consistent across the Product View, Shop Action 5, and Shop Action 9 tags.

3.2 *Category Definition File*

The Category Definition File (CDF) is used to map the Category IDs sent in the tags to a category hierarchy to be displayed in the reports. The CDF is a comma separated values file and contains four columns: (1) the Coremetrics Client ID; (2) Category ID; (3) Category Name; and (4) Parent Category ID.

Every Category ID sent through the tags should have a corresponding line within the CDF that defines its Category Display Name and Parent Category ID. Every Parent Category ID should also have a line in the CDF file that will map it to its Display Name and Parent Category ID. For top-level categories, the Parent Category ID will be left empty. Coremetrics can then recreate the appropriate category hierarchy tree by following the Parent Category ID references up to the top-level categories.

The CDF allows update of the existing category hierarchy without modification of the tag implementation site code. Updated or rearranged category organization can be created in the reporting through modification of the CDF. This simplifies management of the reporting hierarchy.

The CDF should be uploaded to Coremetrics through the CDF import tool at https://import.coremetrics.com. This tool allows direct uploads of CDF's through the gui or creation of an ftp (port 22) or sftp (port 998) server account with Coremetrics (ftp.coremetrics.com) for scheduled upload of automatically generated CDF's. CDF updates are effective in reporting from the date of upload forward: historical categorization is not affected by upload of an updated CDF. The frequency of CDF generation and upload should be based on the frequency of changes to your Page and Product categorization hierarchies. Coremetrics recommends creating an automated script generating the CDF and uploading at the desired frequency, up to once per day, using your ftp or sftp account managed through https://import.coremetrics.com.

For more details about the creation and formatting of the Category Definition File, see Appendix 8.1.

3.3 *Product Categories Categorization Inheritance*

In order to simplify implementation of Product tag categorization, Coremetrics features 'server-side' Category ID inheritance processing. These rules are applied during processing of daily report data. This processing will cause uncategorized Shop Action 5 and Shop Action 9 tags to be categorized according to other categorized Product View tags or Shop Action tags collected for the same Product within the same data collection 'session'.

3.3.1 Category Inheritance Rules

Inheritance of Categorization at time of report processing follows these rules:

1. **Product View** tags do not inherit categorization from other tags including other categorized Product View tags in the same session. The Product View tag should always be collected with a valid Category ID parameter.

2. **Shop Action 5** tags collected without a Category ID will inherit the Category ID from another same-session Shop Action 5 tag having the same Product ID. If no matching Shop Action 5 tag with a categoryID is found in the session, inheritance will devolve upon a matching Product View tag with a non-null Category ID.

3. **Shop Action 9** tags collected without a Category ID will inherit the categoryID from another same-session Shop Action 9 tag having the same Product ID. If no Shop Action 9 tag with a Category ID is found in the session for that Product ID, inheritance will devolve upon the Shop Action 5 tag or Product View tag, in this order: 1) a matching Shop Action 5 with a non-null categoryID or 2) a matching Product View tag with a non-null Category ID.

3.3.2 Exceptions

In certain special cases Coremetrics categorization inheritance processing may not be able to achieve 100% complete categorization in Product Categories.

1. **Persistent Carts:** visitors viewing a saved cart in a new session, and/or completing a purchase may not view product detail pages, resulting in lack of a categorized Product View tag for the Shop Action tags to inherit categorization from. In this case, a valid Category ID parameter value should be included with the Shop Action 5 tags sent when the persisted cart is retrieved and viewed. Shop Action 9 tags will inherit categorization from the Shop Action 5 tags and need not be categorized.

2. **Direct add-to-cart Site Functionality**: This site functionality typically allows visitors to bypass the product detail page and associated Product View tag data collection by adding items directly to the cart from product category display pages. If no Category ID value is sent with a Shop Action 5 tag, this tag and any subsequent Shop Action 9 tags for this Product ID will be uncategorized in reporting due to lack of a categorized Product View tag in the session. In this case, a valid Category ID parameter value should be included with the individual Shop Action 5 tag sent when the add-to-cart event occurs.

4 Marketing Link Tracking

Coremetrics can track the performance of both offsite and onsite Marketing links through the use of URL query string parameters. Specific query string parameters are used to indicate the type of marketing program and link that is tracked:

- Marketing Management Center Reports (MMC)
- Site Promotions Reports (SP)
- Real Estate Analysis Reports (RE)

It is recommended that marketing link tracking be implemented after initial data collection implementation is complete and verified. This information is provided to facilitate planning for implementation of web and email-based marketing link tracking. Coremetrics will provide additional best practice guidance on this subject during post-implementation report training and applications enablement.

Please reference the Coremetrics Online Analytics (http://welcome.coremetrics.com) Marketing tab / 'Tracking Code Generator' utility for generation of MMC, RE or SP parameters. The 'Tracking Code Generator' allows upload of bulk sheets of marketing URL's, generation and appending of parameter values to the URL's, and download of finished URL's for delivery to partners or internal marketing departments.

- ***Hash values in marketing link URLs***

Destination links incorporating '#' characters into the page or link URL prior to the cm_mmc=, cm_sp= or cm_re= parameters will prevent the specified Marketing Program, Site Promotion or Real Estate values from appearing in Coremetrics reporting: Coremetrics ignores marketing parameters found after any '#' character in page URL or anchor link HREF/URL values. Please include Coremetrics marketing tracking parameters prior to '#' characters in any offsite marketing links, final marketing landing page destination URL's, or anchor link HREF/URL values.

4.1 *Offsite Marketing Links*

External marketing campaigns, such as paid search and promotional emails, are tracked through the Marketing Management Center (cm_mmc=) parameter, populating the Marketing tab / 'Marketing Programs' report. By appending a cm_mmc= parameter to the query string of the off-site links to be tracked, Coremetrics will automatically attribute session activity to the related marketing programs.

The cm_mmc= parameter must be found in the destination URL of the first Coremetrics Page View-type tag (pc="Y") collected in the visitor session in order for the session activity to be attributed to the campaign defined in the parameter. This Page View tag must be the first tag received in the visitor session. No special JavaScript tagging is required on the page; however, the Coremetrics libraries and a Page View-equivalent tag must be present on the marketing landing page.

4.1.1 IMAP partners

Tracked offsite marketing links can be generated automatically if your marketing partner participates in the Coremetrics IMAP program. IMAP partners can embed marketing programs deployed through their systems with Coremetrics cm_mmc= parameters to enable automatic tracking and measurement in the Coremetrics Marketing Management Center. Please contact your marketing partner or Coremetrics Support to determine if your partner is an IMAP vendor.

4.1.2 MMC Implementation

The cm_mmc= parameter allows specification of Vendor, Category, Placement, and Item level information. This 'VCPI' information is reported in hierarchical format by Items within Placement, Placements within Category, and Categories within each Vendor. Each of the four values is placed within the cm_mmc= parameter, delimited by the value "-_-". Spaces between words are represented by %20.

Example: http://www.client.com/product.asp**?cm_mmc=google-_-general-_-swimwear-_-men's%20swimwear**

- Marketing Program report hierarchy:

1) Vendor: google
2) Category: general
3) Placement: swimwear
4) Item: men's swimwear

4.1.3 Manual MMC

Marketing Programs may also be tracked through setting the following page variable on a marketing landing page prior to calling the Page View tag function for that page:

var manual_cm_mmc="vendor-_-category-_-placement-_-item";

The value set in manual_cm_mmc will be automatically applied to the URL collected with the Page View tag from the marketing landing page. This provides a convenient mechanism for implementing code to translate pre-existing marketing link query string parameters into equivalent or similar Coremetrics cm_mmc= tracking codes.

4.1.4 Marketing Attributes for Explore

Up to 50 additional 'Marketing Attributes' may be appended to the tracked marketing URL, in addition to the required cm_mmc= query string parameter. The query string parameter format is cm_mmca# where # is a value "1" through "50".

...&**cm_mmca1**=300x250&**cm_mmca2**=german&**cm_mmca3=**...

The marketing attribute parameters can be included in any order, before or after the 'cm_mmc=' Marketing Program parameter in the marketing URL. Marketing attributes will also work in conjunction with use of the page variable manual_cm_mmc="..." method for implementing Marketing Program tracking.

Collected marketing attribute data will be available for inclusion in new Explore Marketing reports.

4.2 *Onsite Marketing Links*

Tracking onsite links through Coremetrics is implemented using the Site Promotions or Real Estate tracking code and associated reports. The type of link to be tracked will dictate which tracking code and report is implemented. Both Site Promotions and Real Estate tracking require appending URL parameters (cm_sp= or cm_re=) to the HREF attribute of the anchor tag to be tracked. Site Promotions and Real Estate tracking can be applied indivually or in combination to the same link. Example: ?cm_re=1-_-2-_-3&cm_sp=1-_-2-_-3.

4.2.1 Site Promotions

The Site Promotions report can be used to track the performance of a link across multiple pages. This can be useful for tracking the same promotion running on multiple pages across the site. Links are tracked by adding a cm_sp parameter to the URL query string.

The cm_sp= parameter allows specification of Promotion Type, Promotion and Link level information. This information is reported in hierarchical format by Links within Promotion and Promotions within Promotion Type. Each of these three values is placed within the cm_sp= parameter, delimited by the value "-_-". Spaces between words are represented by %20.

Example: http://www.client.com/product.asp**?cm_sp=fall-_-sale-_-free%20shipping**

- Site Promotions report hierarchy:

  1) Promotion Type: fall
  2) Promotion: sale
  3) Link: free shipping

4.2.2 Real Estate

The Real Estate Analysis report is used for A/B testing or to track the performance of similar links within the same page. Real Estate facilitates performance analysis of the same link hosted in different areas of the same page, or across multiple versions of the same page. Links are tracked by adding a cm_re parameter to the URL query string.

Implementation of Real Estate tracking requires identification of Page ID values for pages which will host links to be tracked. Send the list of Coremetrics Page ID's to Coremetrics Support for configuration. This requirement is unique to Real Estate tracking and is not required for Site Promotion tracking.

The cm_re= parameter allows specification of Version, Page Area and Link level information. This information is reported in hierarchical format by Links within Page-Area and Page-Areas within Version. Each of these three values is placed within the cm_re= parameter, delimited by the value "-_-". Spaces between words are represented by %20.

Example: http://www.client.com/product.asp**?cm_re=page%20A-_-left%20navbar-_-men's %20shirts**

- Real Estate report hierarchy:

  1) Version: page A
  2) Page Area: left navbar
  3) Link: men's shirts

4.2.3 Manual RE/SP

Site Promotions and Real Estate links may also be tracked through setting the following page anchor attribute in the related anchor tag:

…
…

The value set in manual_cm_sp and/or manual_cm_re will be automatically applied to the URL collected with the associated 'Link Click' tag generated automatically upon visitor click of the anchor link. This provides a convenient mechanism for implementing Site Promotions and/or Real Estate tracking programmatically or for Search Engine Optimization purposes.

4.2.4 Automatic Impression Tag Server Call Charges

Real Estate, Site Promotions and WebSphere Commerce E-Marketing Spots will generate impressions as well as clickthroughs. Every 10 impressions captured will cost between 1 and 4 server calls. The exact server calls charged varies based on the total combinations of unique SP, RE or CM parameter values processed. For example, if 100 links in the homepage contain Site Promotions/Real Estate parameters and this page is viewed 500,000 times, this will result in an additional 5 to 20 million server calls charged for that period.

By default, impression data collection for all On-Site Marketing Tracking types is enabled:

- **cm_TrackImpressions="RSCM"**

- **Code Definitions:** R=Real Estate impressions; S=Site Promotion impressions; CM=WebSphere Commerce E-Marketing Spot impressions.

Automatic Impression data collection can be completely or selectively disabled at the page level through the cmSetupOther function:

- **Disable all impressions:** cmSetupOther({"cm_TrackImpressions":""});
- **Enable only Real Estate impressions:** cmSetupOther({"cm_TrackImpressions":"R"});
- **Enable only Site Promotion impressions:** cmSetupOther({"cm_TrackImpressions":"S"});
- **Enable only WebSphere Commerce E-Spot Impressions:** cmSetupOther({"cm_TrackImpressions":"CM"});
- Other combinations are possible, such as “RS”, “RCM”, “SCM”, etc.

For implementations not using Coremetrics Hosted libraries (//libs.coremetrics.com/eluminate.js), impressions are controlled by directly setting var cm_TrackImpressions ="" in the page:

- var cm_TrackImpressions ="";

4.2.5 Manual Linkclick and Impression Tracking

In certain cases the automatic collection of linkclick and/or impression tags used by Site Promotions and Real Estate reporting cannot function. Common examples: 1) anchor tags without HREF= attributes; 2) Flash, Silverlight and other interactive objects without HTML anchors. Additionally, any anchor tags containing invalid HREF= url values such as “javascript:function”, “mailto:address...” will prevent reporting of clicks and impressions for Site Promotions and Real Estate.

In these cases it may be necessary to implement calls to the ‘manual’ equivalents of the impression and linkclick tags. See section 2.5.12-13 for details on the ‘manual’ linkclick and impression tags.

5 Testing Tools

This section describes the test tools available to help in the coding and debugging of a Coremetrics implementation. There are three main tools that can be used during the development process—the Coremetrics TagBar, the Implementation Test Tool (ITT), and Coremetrics Test Reports.

5.1 *Coremetrics TagBar*

5.1.1 Description

The Coremetrics TagBar is an Internet Explorer plug-in that allows you to view all the tags being sent to Coremetrics from a Coremetrics tagged page. The TagBar should be used to ensure that the tags on a page are sending the appropriate values in the appropriate fields. Any tags not showing up in the TagBar could indicate a problem with the code or a JavaScript error that is preventing the tag from being rendered.

5.1.2 Where to find it

The TagBar can be downloaded from https://support.coremetrics.com using your assigned production report logon credentials. TagBar can also be installed as part of the Coremetrics 'Tools' plugin downloadable directly from the Content tab / 'LIVEview' section of http://welcome.coremetrics.com Analytics reporting. Please contact Coremetrics Support or your Coremetrics report administrator to obtain report logon credentials.

5.1.3 How to use it

Once installed, the TagBar can be activated by clicking on the Coremetrics icon  in the Internet Explorer toolbar. This will open the TagBar in the lower portion of the browser. Within the TagBar frame, all Coremetrics tags on the current page will be shown, including all the values set within the tag. The TagBar indicates whether those tags are pointed to the test environment (testdata.coremetrics.com) or the production environment (data.coremetrics.com) in parenthesis next to the name of the tag.

The Action menu in the upper left corner provides a list of actions that can be taken with the TagBar.

- **Refresh Tag Display** – updates the tags displayed in the TagBar.
- **Copy Selected Text** – copies any text that is selected within the TagBar to the clipboard.
- **Show/Copy Page ID –** displays the Page ID for the current page and copies the value to the clipboard.
- **Open Tag Monitor –** opens a new window to display the Tag Monitor, which records all tags sent to Coremetrics in a list form, separated by lines to indicate tags sent from the same page.

5.2 *Implementation Test Tool (ITT)*

5.2.1 Description

The Implementation Test Tool is a web-based interface that allows you to see what data has reached the testdata.coremetrics.com environment. During development of Coremetrics tags, ITT should be used periodically to verify that the data sent in the tags is reaching the Coremetrics test environment in the appropriate format.

Data received through tags will usually show up in ITT within 1-2 minutes of being sent. Only data for the current day will be available, as the data gets rolled off at the end of each day.

5.2.2 Where to find it

ITT can be accessed at http://itt.coremetrics.com. You will be asked to provide a user name and password to access the site. Please contact Coremetrics Support to obtain access.

5.2.3 How to use it

ITT requires you to fill in certain information and choose which set of data you would like to access. The fields that must be entered are:

- **Client ID** – the client specific ID assigned by Coremetrics for the particular site
- **Cookie ID**
  - **My Cookie** will only show activity associated with the computer currently accessing ITT.
  - **All Cookies** will show data for all activity.
  - **Other Cookie** will show data for the specific cookie ID entered in the field.
- **Date range –** the date/time of the data to be accessed. If left blank this will retrieve all data available. *Note*: Only data for the current day is available, since the data gets rolled off every day.
- **Data Type –** specifies the type of data the user is interested in seeing from ITT. Most of these types correspond with specific Coremetrics tags.

Please direct any questions you may have about these input values to Coremetrics Customer Support.

5.3 *Coremetrics Test Reports*

Coremetrics test reports are available upon request from Coremetrics Support. This subset of production reports is used to evaluate test site data collection and reporting prior to production launch. Activation or re-activation of test reporting can be requested from Coremetrics Support.

Test report data retention period is limited to five weeks. Only data up to five weeks in the past can be accessed in the test reporting. Test reports are automatically deactivated and all test data deleted if either of the following conditions are met for 30 consecutive days: 1) no logon to the associated 6-series test ID has occured; 2) no data has been sent to the 6-series test ID.

5.3.1 Description –Test System reports

Client IDs beginning with '6' include test versions of Analytics (welcome.coremetrics.com) and Explore if contracted (explore.coremetrics.com).

- **The following applications are <u>available</u> for Test IDs.**

Analytics (welcome.coremetrics.com); Explore (if contracted for the related production ID, explore.coremetrics.com); Import (import.coremetrics.com) and LIVEview. All reports within these applications are available except as noted below.

- **The following applications and reporting features are <u>not available</u> for Test IDs.**

Applications: Monitor, Intelligent Offer, AdTarget, Export
Analytics: Reports/Custom Reports/DIP, WebSphere Commerce Module
Explore Modules: Registration (reporting), Registration Import, Impression Attribution, Multichannel Import.

5.3.2 Where to find the reports

Coremetrics test reports can be accessed at https://welcome.coremetrics.com using your "6-series" ID – replace the first digit in your Client ID with a "6". Import can be found at https://import.coremetrics.com. Explore (if contracted) can be found at https://explore.coremetrics.com

5.3.3 How to use the reports

In order to access the test reports, you must enter the Coremetrics assigned client ID, replacing the first digit with a "6", a username, and password. Once logged in, you will see the reporting interface.

The Analytics Test ID reports most useful during implementation for confirmation of collected data tags are: Top Line Metrics, Product Categories, Page Categories, On-Site Search, Elements and Events.

In order for the Product and Page Categories reports to map the category IDs to a hierarchy, a Category Definition File (CDF) must be uploaded. You can upload the CDF to test reports by logging in to https://import.coremetrics.com using your 6-series Client ID.

6 First Party Data Collection – 'Client Managed' and 'Coremetrics Managed'

Two forms of first party data collection are available: 'Client Managed' and 'Coremetrics Managed'. Your Coremetrics Sales or Account representative, Provisioning Services, or Support contacts will provide information regarding which solution is specified for your implementation. In most cases 'Client Managed' will be provisioned and the steps in this document required for implementation of 'Coremetrics Managed' become unnecessary.

Implementation of 'Coremetrics Managed' first party data collection involves a certain amount of administrative overhead and recurring costs and maintenance for both Coremetrics and your network team. 'Client Managed' first party on the other hand requires minimal setup effort and

has no recurring cost or administrative overhead. In most cases no additional effort is required when implementing 'Client Managed' first party data collection. To determine which solution will meet your needs please contact your Coremetrics Account or Sales team, or Coremetrics Customer Support.

The remainder of this document describes the 'Coremetrics Managed' first party implementation in detail.

Coremetrics Managed First-party data collection is enabled by configuring Coremetrics data-collection infrastructure as a sub-domain derived from the Clients' own home domain. This sub-domain will reference Coremetrics' Global Load Balancers (GLBs) while being identified via secure certification as a Client-controlled resource. Visitors browsing the Client's web pages will be served a first-party cookie on the Client's behalf by Coremetrics. User browsing information is captured via JavaScript driven browser-requests to the data-collection sub-domain and is associated with the profile via the unique ID stored on the cookie.

6.1 *Coremetrics Managed 1st Party Deployment Process*

6.1.1 Initial Planning

a. Walk-through of proposed DNS changes
b. Discussion of domain name selection for the data-collection server
c. SSL Certificate procurement and maintenance requirements
d. Privacy Policy content, Opt-Out functionality and placement

6.1.2 Determine Sub-Domain Name for Data-Collection

The Client will supply Coremetrics with a sub-domain name of their choice (field 1 of the 'Certificate Signing Request' form). The name need not comply with a particular schema, but Coremetrics recommends that the name is consistent with the web-names that the Client already uses. Coremetrics Support or Implementation Team can advise on names that are most inconspicuous.

Examples for a theoretical Client's domain:

- www3.CLIENTDOMAIN.com
- server2.CLIENTDOMAIN.com
- newton.CLIENTDOMAIN.com

6.1.3 Provide Certificate Signing Request Information

Coremetrics will provide a 'Certificate Signing Request' (CSR) form for completion by Client. This form can be requested from Coremetrics Support or your assigned Implementation Engineer. The completed form should be returned to Coremetrics Support or the assigned Implementation Engineer. The CSR data will be submitted by Coremetrics to the Certifying

Authority (CA) and the generated 'key' file forwarded back to the Client for ordering of SSL Cert. To ensure expediency and avoid rejections, it is critical that the Client gather complete and accurate CSR information before exchange with Coremetrics. CSRs contain fundamental Client information including:

- SSL Data Collection Domain name
- Corporate Technical Contact Information (Name, Phone, Title, Address, Email, FAX)
- Location Information (Address, Country, etc)
- Organizational Information (Company, Department, etc)
- Web Administration information (Contact Names, numbers, login, etc)

Coremetrics Support or your assigned Implementation Engineer will answer questions and assist with this process.

6.1.4 Obtain Secure Sockets Layer (SSL) Certificates

The client will perform certificate acquisition using the Coremetrics-provided key file and forward the received SSL Certificate file to Coremetrics Support or assigned Implementation Engineer for installation on Server Load Balancers at each of the redundant Coremetrics datacenters. Note that when possible, additional licenses should be used in lieu of additional certificates to unify management.

6.1.5 Configure Client-side Name Server

The Client's IT department or the Client's ISP must configure their Domain Name Server(s) (DNS) with NS records to properly refer to the Coremetrics GLBs. This process is not dependent on the presence of the SSL certificates and can be performed in advance of Cert install. While unusual, it is possible that a small, one-time fee will be assessed by an ISP to add the required DNS records. It is important that proper record format be used when configuring the Client's DNS.

- Example DNS 'Name Server' Records for a Data Collection subDomain (dcd):

<your dcd>.sitedomain.com IN NS ns1.coremetrics.com.
<your dcd>.sitedomain.com IN NS ns2.coremetrics.com.
<your dcd>.sitedomain.com IN NS ns3.coremetrics.com.

The NS record TTL values should be set to the DNS provider preference for caching responses to requests from the subdomain (typically this will be a high value).

There should be only three entries within your DNS settings that contain the sub domain delegated to Coremetrics. This domain should be delegated for use solely with Coremetrics data collection – please check that there are no other DNS entries for this sub domain.

Unless your DNS settings complete this automatically, note that the "period" at the end of each entry is required. Depending on the interface used to create the entries, the period may be implicitly included.

6.1.6 Privacy Policy Updates & Implementing Opt-Out

Coremetrics always recommends that clients follow industry best practices and obtain all necessary consents from visitors to the client's web site. In addition, Coremetrics strongly suggests that clients update privacy policy pages to (1) advise visitors of the client's data collection and use practices, (2) notify that cookies are being placed on the user's computer with an explanation of the purpose and utilization of these cookies, and (3) provide an integrated "opt-out" functionality to accommodate user's who choose not to have their browsing data collected. Please see section 7 for further information on Privacy Considerations

6.1.7 Solution Rollout

The final configuration required for first party data collection is to update the cmSetClientID(…) function call to reference the new data collection domain. The Data Collection Domain parameter should now be *<data collection domain>.clientdomain.com* rather than data.coremetrics.com. See section 2.3 'Setting Data Collection Parameters' for more information.

7 Privacy Considerations

Coremetrics highly recommends that clients update their privacy policy to notify visitors of tracking and offer an opportunity for the visitor to opt out of the data capture.

7.1 *Privacy Suggestions*

Coremetrics advises clients to maintain compliance with the core FTC standards for fair information practices: i) Notice; (ii) Choice; (iii) Access, and (iv) Security. This includes providing notice in privacy statements that indicates what data is being collected and how the data is being analyzed, including situations such as described above, where data is integrated from multiple sources. Coremetrics strongly advises Clients to offer their customers choice; that is, the ability for Web visitors to opt-out of having their behavior collected, either by a partial opt-out (to have the visitor behavior data analyzed anonymously) or with a total opt-out (to not collect any visitor experience data). In cases where policies provide statements about choice, Coremetrics' will require Clients using First-party data collection to provide their visitors these privacy services either through an opt-out function, or by providing instructions to visitors on configuring their browsers appropriately. Coremetrics is the only analytics company to provide Clients the ability to deploy a robust, integrated opt-out function free from maintenance overhead. This ensures Client compliance with FTC regulations, and avoids troubling visitors with the inconvenience of deciphering complex instructions. Deploying an opt-out on the Client's Web site also demonstrates to visitors an expected level of respect and proves that the Client is committed to online privacy.

It is important to note that other forms of Web traffic analysis, such as log file-based solutions, cannot offer a full range of choice for a Web site visitor. Since all Web traffic is automatically collected in a log file, Web behavior data is recorded for all visitors. Without an internally-architected custom mechanism for opting out of Web site tracking, or sophisticated filters on data acquisition streams, companies that choose log-file based analytics will not be compliant with the notice and choice guidelines recommended by the FTC. Furthermore, these solutions need to be internally architected so that the visitor identification cookies are P3P compliant, resulting in higher costs and additional internal expertise to support.

In summary, by choosing the Coremetrics solution for online analytics, clients are also choosing to adhere to industry best practices for privacy. Coremetrics and its data collection comply with FTC guidelines for notice and choice and all data capture is fully P3P compliant. In addition to product development cycles to meet new standards, Coremetrics has a Chief Privacy Officer on staff, ensuring that the Coremetrics data collection technology conforms with the most current generally accepted Internet privacy standards and any regulatory framework which may be put in place.

7.2 *Implementing Opt-Out with Coremetrics First Party Data Collection Solutions*

The Coremetrics first-party data collection solution enables Coremetrics clients to add opt-out functionality directly within their privacy policy or other appropriate webpage. Clients adding this capability enable site visitors to directly control their choice of participation in data collection, while removing the burden of basic user-privacy administration. Additionally, giving visitors the ability to execute choice while reviewing data collection and privacy policies is consistent with industry best practices and FTC privacy guidelines. For a more general discussion of this topic, please refer to the "Coremetrics 2007 Privacy Guidelines" brief.

Two forms of opt-out implementation are available for support of '**Coremetrics Managed'** and '**Client Managed'** First Party data collection methods. Sample implementations for each data collection method are provided at the end of this document. To determine your current First Party data collection method and which opt-out solution you should implement, please contact your assigned Coremetrics Implementation Engineer or Coremetrics Customer Support.

7.2.1 Opt-Out Description

There are three levels of data collection possible for the Client to enable: 1) full participation; 2) "opt-out" of data collection, or 3) anonymous participation. Users will be presented with a simple selection of a radio button identifying their choice and then clicking a submit button.

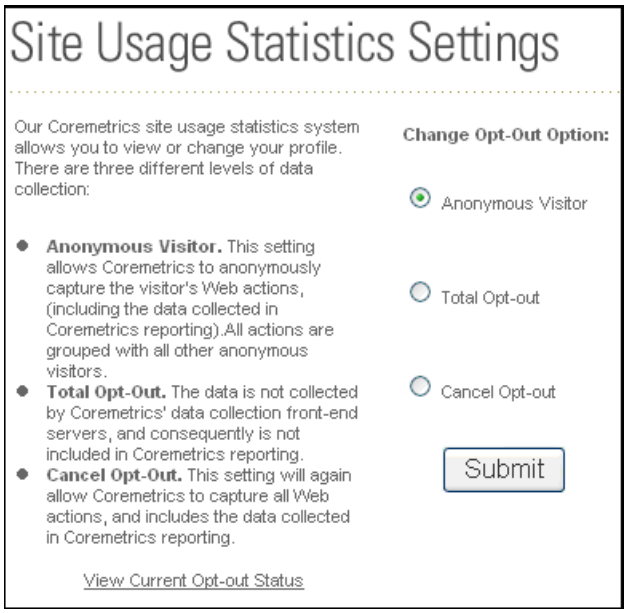

Figure 1: Example implementation of Opt-Out dialog.

Typically, visitors will be presented with a radio button identifying their choice and then clicking a submit button. An additional function to query the status of visitor opt-out is also available.

- **Anonymous Visitor.** This setting allows Coremetrics to anonymously capture the visitor's basic site actions and to include the data in Coremetrics' reporting. All actions are grouped within metrics for all other anonymous visitors.
- **Total Opt-Out.** The data is not collected by Coremetrics' data collection front-end servers, and consequently is not included in Coremetrics' reporting.
- **Cancel Opt-Out.** This setting allows Coremetrics to capture all Web actions and to include the data in Coremetrics' reporting.

7.2.2 General Use-case

The following is an outline description of the process a site visitor may experience when utilizing the Opt-Out functionality:

1. The visitor views the Privacy Policy and Opt-Out Options.
2. The visitor chooses a link to the "Opt-Out options" page.
3. The visitor selects one of the opt-out options, and clicks the "Submit" button.
4. For 'Coremetrics Managed 1st Party' implementations, the opt-out options are transmitted directly to <dataSubdomain.ClientDomain.com> (the Client hostname mapped to Coremetrics). This data transfer and cookie update are transparent to the visitor experience.
5. For 'Client Managed 1st Party' implementations, the opt-out options cookie update is performed by the Coremetrics data collection library included in the site page. The cookie update is transparent to the visitor experience.
6. A small pop-up window is displayed to the visitor confirming the execution of the selected optout option.

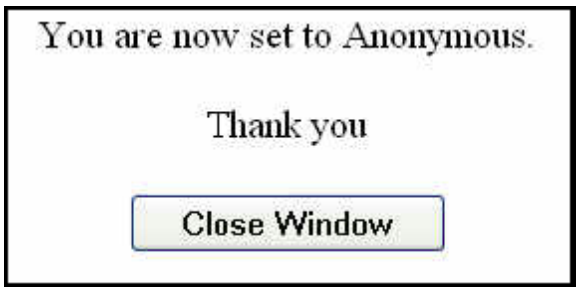

Figure 2: Example confirmation.

7. The visitor closes the pop-up window, and then continues to browse the site, confident that he/she was able to choose the appropriate opt-out/opt-in functionality.

7.2.3 Implementation – Opt-Out for Coremetrics Managed First Party

To provide opt-out for visitors to a Coremetrics Managed First Party implementation, provide an 'Opt-Out' form or other html allowing the visitor to select opt-out preferences. To avoid sending visitors to another Web site and to maintain the website user interface "look and feel", you may choose to implement the opt-out functionality on your own pages, or as a stand alone page or "pop-up." The implementation of the opt-out functionality is typically accomplished with the provision of the appropriate descriptive language in the site Privacy Policy page and creation of an opt-out HTML form.

The Coremetrics Managed First Party opt-out HTML form is described below:

Opt-Out HTML Form

The HTML code below exemplifies the implementation of the Opt-Out form. Note the functions in the <head> that are required in order to provide for the form functionality. A client may also specify the background color or image of the pop-up windows that are displayed to the user.

This code should be included into a page on the client website with appropriate formatting, images, etc. to integrate the view.

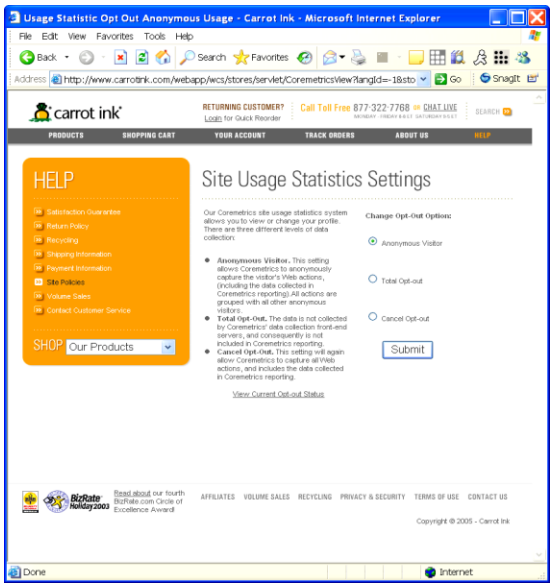

Figure 3: Example in-page opt-out functionality supporting a Coremetrics Managed First Party Implementation.

Notes:

- This code should be attached as file: **"opt_out.html"**.
- **"DA_Sub-Domain.ClientDomain.com"** should be replaced with your assigned Coremetrics Managed First Party Data Collection Domain (for example: "http://**ww12.yoursite.com**/privacy/getStatus.php").

```
<html>
<head>
<title>Anonymous and Optout page</title>
<script language="JavaScript">
<!--
var newWindow;
function viewStatusWindow () {
bg_color="FFFFFF"; // optional background color of the popup window
bg_img=""; // optional background image for the popup window
// complete url needed
newWindow=window.open
("http://DA_Sub-Domain.ClientDomain.com/privacy/getStatus.php"+
"?bg=" + bg_color + "&im=" + bg_img, "popup1",
"resizeable,width=500,height=400")
}
function optResultWindow ( f ) {
// destination_opt_out - url of the page that is displayed in
// the pop up window after the opt-out cookie is set
destination_opt_out="http://DA_Sub-Domain.ClientDomain.com/privacy/optout.html";
// destination_anonymous - url of the page that is displayed in
// the pop up window after the anonymous cookie is set
destination_anonymous="http://DA_Sub-Domain.ClientDomain.com/privacy/anonymous.html";
// destination_cancel - url of the page that is displayed in
// the pop up window after the cancel cookie is set
destination_cancel="http://DA_Sub-Domain.ClientDomain.com/privacy/cancel.html";
if ( f.action[0].checked ) {
ac = "anonymous";
destination = destination_anonymous;
```

```
}
if ( f.action[1].checked ) {
ac = "opt_out";
destination = destination_opt_out;
}
if ( f.action[2].checked ) {
ac = "optin";
destination = destination_cancel;
}
newWindow=window.open (
"http://DA_Sub-Domain.ClientDomain.com/privacy/privacy_handler.php"+
"?dest=" + destination + "&act=" + ac,
"popup1", "resizeable,width=500,height=400")
}
function setStatus(msg) {
status = msg
return true
}
//-->
</script>
</head>

<body>
<p><a href="javascript:void(0)" onClick="viewStatusWindow()"
onMouseOver="return setStatus('Click to view Opt-out status')"
onMouseOut="return setStatus('')"><u>View Current Opt-out
Status</u></a></p>
<hr>
<table width="595" border="0" cellspacing="0" cellpadding="1">
<tr>

<td>
<form name="optout">
<p><font face="Arial, Helvetica, sans-serif" size="2"><br>
<b>Opt-out Selection Form:</b></font><br><br><br>
<font face="Arial, Helvetica, sans-serif" size="2">
<input type="radio" name="action" value=""anonymous" checked>
<b>Anonymous Visitor.</b> </font></p>
<p><font face="Arial, Helvetica, sans-serif" size="2">
<input type="radio" name="action" value="opt_out" >
<b>Total Opt-out.</b> </font></p>
<p><font face="Arial, Helvetica, sans-serif" size="2">
<input type="radio" name="action" value="opt_in">
<b>Cancel Opt-out.</b> </font></p>
<p>
<input type="button" value="Submit"
onClick="optResultWindow(this.form)"></p>
</form>
</td>
</tr>
</table>
</body>
</html>
```

7.2.4 Customizing the Coremetrics Managed 1st Party Opt-Out HTML Response and Status Windows

- **Customizing the Response Window**

It may be desirable to replace the standard Coremetrics Managed 1st Party opt-out response window content with localized language or other customized content. See Figure 2., above, for an example of the standard response window for anonymous opt-out.

In order to replace the Coremetrics default window content, simply set the "**destination**" parameter in the opt-out code to the URL hosting your custom content. This content should be suitable for display in a resizable width=500/height=400 window unless you plan on changing the initial window size.

```
if ( f.action[0].checked ) {
ac = "anonymous";
destination = "http://www.mysite.com/customcontent_anonymous.html";
}
if ( f.action[1].checked ) {
ac = "opt_out";
destination = "http://www.mysite.com/customcontent_opt_out.html";
}
if ( f.action[2].checked ) {
ac = "optin";
destination = "http://www.mysite.com/customcontent_cancel.html";
}

newWindow=window.open (
"http://DA_Sub-Domain.ClientDomain.com/privacy/privacy_handler.php"+
"?dest=" + destination + "&act=" + ac,
"popup1", "resizeable,width=500,height=400")
```

- **Customizing the Status Window**

The language presented in the standard Coremetrics Managed 1st Party opt-out getStatus window content can be presented in a number of languages, controlled by the optional ?lang=xx parameter. The default language is English. The following example illustrates changing the default status language to German.

```
newWindow=window.open
("http://DA_Sub-Domain.ClientDomain.com/privacy/getStatus.php"+
"?lang=de" + "&bg=" + bg_color + "&im=" + bg_img, "popup1",
"resizeable,width=500,height=400")
}
```

Supported language codes: en, fr, es, de, cn, da, nl, fi, it, ja, ko, pt, sv

7.2.5 Implementation – Opt-Out for Client Managed First Party

To provide opt-out functionality for visitors to a Client Managed First Party implementation, provide an 'Opt-Out' form or other html allowing the visitor to select opt-out preferences. The opt-out HTML should call the function SetOptOut(value); with one of three possible parameter values: 1) empty ('') for opting-in to full data collection; 2) 'opt_out' for complete opt-out of data collection; 3) or 'anonymous' for opting-in to anonymous data collection.

Opt-Out HTML Example

The HTML code below provides an example opt-out page submitting the SetOptOut function in response to visitor choice. Please modify this HTML to incorporate seamlessly into your website with appropriate language, navigation, formatting and images.

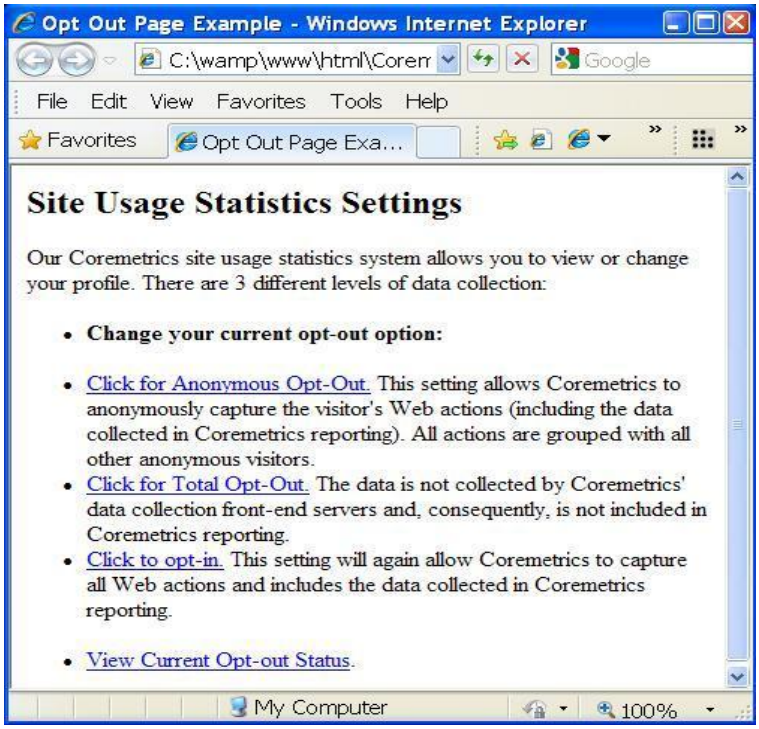

Figure 4: Example in-page opt-out functionality supporting a Client-Managed First Party Implementation.

```
<HTML>
<HEAD>
<TITLE> Opt Out Page Example </TITLE>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
// send data to production - Client-Managed 1st Party
cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
</HEAD>
<BODY>
<script type="text/javascript">
var currentStatus = cI("CMOptout");

if (!currentStatus) {
      currentStatus = "opt-in";
}
function setOptOut(value) {
      var futureDate = new Date();
      futureDate.setFullYear(futureDate.getFullYear() + 20);
      document.cookie = "CMOptout=" + value + "; expires=" +
futureDate.toGMTString();
      currentStatus = cI("CMOptout");
      if (!currentStatus) {
            currentStatus = "opt-in";
```

```
      }
}
</script>

<div id="customerServ-header">
<h2>Site Usage Statistics Settings</h2>
</div>

<div id="browse-categories" class="clearfix">

<p>Our Coremetrics site usage statistics system allows you to view or change your
profile.  There are 3 different levels of data collection:</p>

<div id="opt-out-description">
<ul>
<li><h4>Change your current opt-out option:</h4></li>

<li><a href="#opted-out-anonymous" onclick="setOptOut('anonymous');">Click for
Anonymous Opt-Out.</a> This setting allows Coremetrics to anonymously capture the
visitor's Web actions (including the data collected in Coremetrics reporting). All
actions are grouped with all other anonymous visitors.</li>

<li><a href="#opted-out-total" onclick="setOptOut('opt_out');">Click for Total Opt-
Out.</a> The data is not collected by Coremetrics' data collection front-end
servers and, consequently, is not included in Coremetrics reporting.</li>

<li><a href="#opted-in" onclick="setOptOut('');">Click to opt-in.</a> This setting
will again allow Coremetrics to capture all Web actions and includes the data
collected in Coremetrics reporting.</li>

<br><br>

<li><a href="#check-status" onclick="alert('Your current status is: ' +
currentStatus);">View Current Opt-out Status</a>.</li>
</ul>
</div>
</div>

<script type="text/javascript">
cmCreatePageviewTag( "OPT-OUT PAGE", "PRIVACY");
</script>

</BODY>
</HTML>
```

8 Appendices

8.1 *Category Definition File*

8.1.1 Introduction

In order to define the category structure and display names for the category IDs that you will be sending through the tags, you must use a Category Definition File (CDF). This file maps each category ID sent in the tags to a display name and a parent category. The file should be uploaded

to Coremetrics through the https://import.coremetrics.com Import GUI or the FTP account configured and managed through the GUI.

8.1.2 Category Definition File Format

The Category Definition File is a comma separated values (CSV) file. Each line describes a category in the hierarchy. Each line has four values: Client ID, Category ID, Category Name, and Parent Category ID.

Column	Description
Client ID	Coremetrics assigned ID. This value should be the same for all rows within a given file.
Category ID	Category ID for the category being defined. This should match the values being sent in the Coremetrics tags or referenced in the Parent Category ID column of the CDF.
Category Name	Display Name that should be used for this category ID in the reporting
Parent Category ID	Category ID of this category's parent category. For top-level categories, this column should be empty.

Every category ID that is sent through the Coremetrics tags on your site or referenced in the Parent Category ID column should have a matching line within the CDF that will map the ID to a display name and parent category. Category IDs that do not have a matching line within the CDF will be lumped into the "No Category Assigned" top-level category within the reporting.

The Coremetrics TagBar can be used to identify what category ID is being sent in a tag. The Site Location ID (cg) field within the tags corresponds to the category ID that should match a line in the CDF.

8.1.3 Invalid Characters

Single Quotes (') and Double Quotes (") are automatically removed from collected tag Category ID parameter value and thus no tag data can be assigned to Category ID's including quotes.
Single and Double quotes may be included in Category Name column values for report display purposes. This is accomplished by enclosing the Category Name in double quotes:

99999999,204,"DRILL BITS 1/4" TITANIUM",202

Commas (,) are removed from all collected tag data prior to reporting (replaced with a space), including tag Category ID's. For this reason, commas should not be included in CDF values.

8.1.4 Multibyte CDF characters

Only singlebyte data should be included in the CDF supporting a singlebyte-enabled Client ID. Categories including multibyte characters in the CDF for a singlebyte Client ID will not be available in reporting.

CDF column data (Category ID, Name and Parent) for multibyte-enabled Client ID's may include multibyte characters.

8.1.5 Example file

Below is an example category structure with the corresponding CDF. Please note that all text in the file should be in all upper case.

Category Structure

Client ID	Category ID	Category Name	Parent Category ID
99999999	101	Mens	
99999999	201	Womens	
99999999	301	Sale	
99999999	102	Mens Shirts	101
99999999	103	Mens Pants	101
99999999	104	Mens Shirts: Dress	102
99999999	105	Mens Shirts: Tee	102
99999999	202	Womens Shirts	201
99999999	203	Womens Pants	201
99999999	204	Womens Shirts: Dress	202
99999999	205	Womens Shirts: Tee	202
99999999	302	Sale: Shirts	301
99999999	303	Sale: Brands	301

Example File

99999999,101,MENS,
99999999,201,WOMENS,
99999999,301,SALE,
99999999,102,MENS SHIRTS,101
99999999,103,MENS PANTS,101
99999999,104,MENS SHIRTS: DRESS,102
99999999,105,MENS SHIRTS: TEE,102
99999999,202,WOMENS SHIRTS,201
99999999,203,WOMENS PANTS,201
99999999,204,WOMENS SHIRTS: DRESS,202
99999999,205,WOMENS SHIRTS: TEE,202
99999999,302,SALE: SHIRTS,301
99999999,303,SALE: BRANDS,301

8.1.6 Uploading the CDF

The CDF should be uploaded to Coremetrics through the CDF import tool at https://import.coremetrics.com. This tool allows direct uploads of CDF's through the gui or creation of an ftp (port 22) or sftp (port 998) server account with Coremetrics (ftp.coremetrics.com) for scheduled upload of automatically generated CDF's.

Files imported through the Import GUI or sent to ftp.coremetrics.com will be automatically imported and processed, updating your categorization hierarchy in reporting from that point forward. Changes from an uploaded file should be reflected in the next processing of the daily reports. The frequency at which the CDF is uploaded can be determined based on your own needs. For clients whose hierarchies may change on a daily or weekly basis, Coremetrics recommends setting up an automated script that will generate the CDF and upload it daily. This will ensure that Coremetrics picks up any changes in the hierarchy on a daily basis.

8.1.7 File Naming Convention

The CDF should be named according to the following convention:

CDF_<client_id>.csv

Where **<client_id>** is your Coremetrics assigned client ID.

8.2 *Data Integrity Process File*

8.2.1 Introduction

Coremetrics uses the Data Integrity Process (DIP) to validate the data being collected by the Coremetrics tagging. DIP compares data directly imported from your backend database to the data collected and used by Coremetrics reporting. Data is imported through the upload of a DIP file, which is sent via FTP to Coremetrics on a daily basis. This feature is not available in Test reporting.

8.2.2 DIP File Format

The DIP file is a comma separated values (CSV) file that contains line item data for all orders placed for the day. Each line has six values: Order Date, Order ID, Product ID, Order Subtotal, Quantity, and Unit Price.

Column	Description
Order Date	Date of the order in appropriate format (see Section 8.2.3)
Order ID	Order ID should match the Order ID being sent in the Coremetrics Order Tag.
Product ID	Product ID for the line item. This should match the Product ID being sent in the Shop 9 Tag.
Order Subtotal	Subtotal for the order. This should not include shipping and handling or tax.
Quantity	Quantity of the product purchased for this line item.
Unit Price	Unit price for the line item.

No quotes should exist anywhere within the DIP file. The DIP file should contain data for all orders placed online that would be tracked with Coremetrics. It should not include any orders placed from other channels, such as store, catalog, or call center/telephone sales. Do not include commas in Order Subtotal or Unit Price values.

8.2.3 Date Formatting

The Order Date should be in the same time zone as your Coremetrics reports were set up to. They should *not* be translated to CST. If you have any questions about what time zone is appropriate, please contact your Implementation Engineer. The Order Date field must be in one of the following formats:

```
DD-MON-YYYY HH24:MI:SS
YYYY-MM-DD HH24:MI:SS
MM/DD/YY HH24:MI:SS
MM/DD/YYYY HH24:MI
DD-MON-YYYY HH24:MI:SS
MM/DD/YYYY HH24:MI:SS
DD-MON-YYYY
MM-DD-YYYY HH24:MI:SS
```

8.2.4 Example File

Below is an example set of order data for one day and the corresponding DIP file entries.

Order Date	Order ID	Product ID	Order Total	Qty	Unit Price
27-Oct-2010 13:32:17	5328031	47175	20.97	3	6.99
27-Oct-2010 14:12:05	5328032	156564	243.87	4	3.49
27-Oct-2010 14:30:07	5328032	187167	243.87	1	29.99
27-Oct-2010 14:55:46	5328032	185767	243.87	4	19.99
27-Oct-2010 14:59:43	5328032	186891	243.87	4	29.99
27-Oct-2010 15:10:23	5328034	188353	231.76	1	19.97
27-Oct-2010 15:19:53	5328034	188396	231.76	2	15.97
27-Oct-2010 15:24:31	5328034	188412	231.76	1	24.97
27-Oct-2010 15:43:14	5328034	188487	231.76	1	29.97
27-Oct-2010 16:03:28	5328034	188529	231.76	1	64.97

Example DIP File

```
27-Oct-2010 13:32:17,5328031,47175,20.97,3,6.99
27-Oct-2010 14:12:05,5328032,156564,243.87,4,3.49
27-Oct-2010 14:30:07,5328032,187167,243.87,1,29.99
27-Oct-2010 14:55:46,5328032,185767,243.87,4,19.99
27-Oct-2010 14:59:43,5328032,186891,243.87,4,29.99
27-Oct-2010 15:10:23,5328034,188353,231.76,1,19.97
27-Oct-2010 15:19:53,5328034,188396,231.76,2,15.97
27-Oct-2010 15:24:31,5328034,188412,231.76,1,24.97
27-Oct-2010 15:43:14,5328034,188487,231.76,1,29.97
```

```
27-Oct-2010 16:03:28,5328034,188529,231.76,1,64.97
```

8.2.5 Uploading the DIP File

The DIP file should be uploaded to Coremetrics' FTP server at ftp.coremetrics.com. Your Implementation Engineer will set you up with an account on the server during the implementation process.

8.2.6 File Naming Convention

The DIP file should be named according to the following convention:

DIP_<client_id>_<YYYYMMDD>.csv

Where **<client_id>** is your Coremetrics assigned client ID and **<YYYYMMDD>** is the date in 'YYYYMMDD' format.

8.3 *Multi-Currency Support*

Coremetrics supports capturing multiple different currencies for purchases under a single client ID. In order to enable this functionality, a currency code that confirms to the ISO4217 specification must be specified. See http://www.xe.com/iso4217.htm for reference.

Currency values must be captured in Shop Action 5, Shop Action 9, and Order tags. In order to capture Currency code with the relevant tags, call the cmSetupOther(...) function with the variable name 'cm_currencyCode' and desired 3-byte currency code:

```
cmSetupOther({"cm_currencyCode":"EUR"});
```

This function can be called in the same script block as the cmSetClientID(...) call on the page.

If not using Coremetrics hosted libraries (//libs.coremetrics.com/eluminate.js) version 4.7.5 or later, the 'cmSetupOther' function will not be available. In this case, call the following function anywhere in the tagged page prior to the related Shop or Order tag function calls:

```
<script>
cmSetCurrencyCode("EUR");
</script>
```

8.4 *Additional Tag Attributes for Coremetrics Explore™*

Coremetrics Explore allows for reporting based on attributes of a tag (e.g. brand, language, author, etc…). Collecting and using attributes in reporting requires the Explore product and the Coremetrics-hosted library (//libs.coremetrics.com/eluminate.js).

The following tag types each support up to 50 attributes with a length of 100 characters per attribute:

Pageview

Productview
Shop Action
Order
Conversion Event
Element
Registration (1-15 are available Analytics, 1-50 are available in Explore reporting)
Marketing Attributes (query string parameters cm_mmca1-50)

Microsoft Internet Explorer request length is limited to 2083 bytes. Please check that your total attribute length for each tag is no more than 1.5k to allow for collection of standard tag parameters (Page ID, category ID, Product ID, etc.), destination and referring URL values, and around 200 additional bytes for data included in all requests.

To obtain Explore please contact your Coremetrics Account or Sales Representative. Please see the Coremetrics Explore User Guide for more information on suggested attributes for capture and how to create reports using attributes.

8.4.1 Capturing Coremetrics Explore™ Attributes

Explore Attribute values are sent to Coremetrics as a single "-_-" delimited tag parameter value. The Explore Attribute tag parameter is located in the parameter list for each tag supporting Explore data collection (see Section 2.5).

- Example "Page View" Tag Function Call with Explore Attributes:

In this example Pageview tag, we are sending the parameters "PageID", "PageCategoryID", "attribute-1", "attribute-3" and "attribute-4". Search String and Search Results will specify JavaScript null values in order to maintain correct parameter order. By not specifying any value for Attribute position 2 in the "-_-" concatenated Attribute string, we retain the correct parameter order for the "attribute-3" and "attribute-4" values.

- Function Definition from //libs.coremetrics.com/eluminate.js:

```
function cmCreatePageviewTag(pageID, categoryID, searchString, searchResults, attributes,
extraFields)
```

- Function Call

```
<script type="text/javascript">
cmCreatePageviewTag("PageID","PageCategoryID",null,null,"attribute-1-_--_-attribute-3-_-
attribute-4");
</script>
```

- Tagbar Test Output:

Page View tag (Test)
Tag Type (tid):"1" (Page View tag)
Page ID (pi):"PageID"
Category ID (cg):"PageCategoryID"

```
Attribute 1 (Explore) (pv_a1):"attribute-1"
Attribute 3 (Explore) (pv_a3):"attribute-3"
Attribute 4 (Explore) (pv_a4):"attribute-4"
```

8.4.2 Video Tracking - Coremetrics Explore™ Attributes

Advanced Explore video tracking can be implemented through the Element Tag 'attributes' 13, 14 and 15:

- **Element ID:** pass the name of the video (e.g. Six Minute Abs)
- **Element Category:** pass the category of the video (e.g. Fitness Videos)
- **Element Attribute Field 13 (e_a13):** Pass the "Video Status:" "0"=Launch; "1"=Pause; "2"=Play;"3"=Completion. Video abandonment/completion rates and average video play times are calculated using the 'Launch' and 'Completion' events. 'Pause' and 'Play' events are sent only in response to clicks on video player 'pause' or 'play' controls. Alias Element Attribute 13 in reporting as "Video Status."
- **Element Attribute Field 14 (e_a14):** Pass the "Video Time Stamp" (in seconds) of the status action. For example, if the user Stops the video 1:23 to the video, pass "83". Alias Element Attribute 14 in reporting as "Video Time Stamp." Video timestamp must be sent for all Video Status values including 'Completion', in which case the value should be equal to the Element attribute 15 'Video Length' value.
- **Element Attribute Field 15 (e_a15):** Pass the "Video Length" (in seconds) of the total length of the video. For example, if the video is 3:10 in length, pass "190". Alias Element Attribute 15 in reporting as "Video Length."

- **Example "Element" Tag Function Calls with Video Explore Attributes:**

In this example sequence we are tracking the video "Six Minute Abs" in category "Fitness Videos" through a hypothetical typical launch, pause, play and completion sequence.

- Function Definition from //libs.coremetrics.com/eluminate.js:

```
cmCreateElementTag(elementID, elementCategory, attributes)
```

- **Step 1:** page or player hosting the video loads. In this example, the video starts playing only when the 'play' control is subsequently clicked by the visitor.

```
cmCreateElementTag("SIX MINUTE ABS","FITNESS VIDEOS","-_--_--_--_--_--_--_--_--_--_--_--_--_-0-_-0-_-190");
```

- **Step 2:** Visitor clicks the 'play' control, starting play of the video – note that the 'play' event need only be sent when a 'play' control is clicked: a video which begins playing at 'launch' of the video does not require collection of an additional 'play' video status event.

```
cmCreateElementTag("SIX MINUTE ABS","FITNESS VIDEOS","-_--_--_--_--_--_--_--_--_--_--_--_--_-2-_-0-_-190");
```

- **Step 3**: Visitor 'pauses' the video halfway through (95 seconds).

```
cmCreateElementTag("SIX MINUTE ABS","FITNESS VIDEOS","-_--_--_--_--_--_--_--_--_--_--_--_-1-_-95-_-
190");
```

- **Step 4:** Visitor resumes play at 95 seconds

```
cmCreateElementTag("SIX MINUTE ABS","FITNESS VIDEOS","-_--_--_--_--_--_--_--_--_--_--_--_-2-_-95-_-
190");
```

- **Step 5:** Visitor watches video to completion – note that timestamp (14) and video length attributes (15) are now equal

```
cmCreateElementTag("SIX MINUTE ABS","FITNESS VIDEOS","-_--_--_--_--_--_--_--_--_--_--_--_-3-_-190-_-
190");
```

8.5 *Additional Product Attributes for Intelligent Offer*

Intelligent Offer supports rules based on 15 static attributes for Products. The product attribute data is uploaded to Coremetrics through the ECDF and EPCMF .csv files documented in section 8.9.1-8.9.2 (Enterprise Product Report).

8.6 *Real-Time Media Tagging*

Coremetrics Monitor now supports two new modules designed to allow real-time analysis of published content. This module requires use of the Coremetrics-hosted library (//libs.coremetrics.com/eluminate.js).

To track designated pages using the Real-Time Media Modules, two additional values must be passed as attributes on the pageview tag. The data format for attributes is described in Section 2.5.1. Note that the Explore product is not required for Real-Time Media, but Coremetrics Monitor is required. The additional values must be passed either in Pageview attributes 1 and 2 or 14 and 15. The collected attributes must be paired together as 1 and 2 or 14 and 15. Failure to specify the values properly may result in incorrect or missing reporting.

The value of attribute 1 or 14 specifies the page of the article (first page, middle pages, last page, or single-page-article). Only one of these values will be set for a given Pageview tag. These values are used to indicate the following:

cm_md_f – indicates the first page of a multi-page article
cm_md_m – indicates a middle page of a multi-page article (i.e. more than 2 pages)
cm_md_l – indicates the last page of a multi-page article
cm_md_fl – indicates a single page article

Important Note: If the article page values are not set correctly (using one of the above 4 values), no data will appear in the Real-Time Media modules.

The value of attribute 2 or 15 specifies the "Article ID", uniqely identifying this article in reporting. The value must be consistent across all pages of a given article. In the case of a single page article, the 'Article ID' value will typically be the same as the Page View tag 'Page ID' parameter value. For multi-page articles, the 'Article ID' will usually be that part of the 'Page ID' which uniquely identifies the article, less any variable data such as page numbers, subtitles, etc.

- Example of the Page View tag attribute values sent when viewing page 4 of a 5 page article, capturing the real-time media values using attributes 14/15:

```
cmCreatePageviewTag("ARTICLE 1: PAGE 4","ARTICLES", null, null, "-_--_--_--_--_--_--_--_--_--_--_--_-
cm_md_m-_-ARTICLE 1");
```

8.7 *Coremetrics Cookies - Technical Appendix*

Coremetrics cookie logic will not interfere with setting of existing non-Coremetrics cookies. Prior to setting cookies Coremetrics checks to determine if such cookies would cause the maximum cookie length or number limit for that browser to be exceeded. In such cases Coremetrics libraries will not set additional cookies, preventing loss of other non-Coremetrics cookies previously set under the domain. Here are the standard Coremetrics cookies set for most implementations:

- Session cookies (3-4 total cookies)
    - 'Session' ID cookie and session-expiration cookie, where '99999999' is the assigned Coremetrics Client ID). Expiration: at end of session.
    - **New Sessions:** Current popular browsers do not begin a new Coremetrics 'session' when a new window or new tab is opened: the existing session set under the specified Data Collection Domain (if Coremetrics Managed) or the Cookie Domain (if Client Managed) will continue to be associated with the current 'session'.
    - **Session Inactivity Timeout:** The default inactivity timeout for Coremetrics 'session' cookies is 30 minutes. If no tags are received for a period of greater than 30 minutes, the next tag received associated with this session will cause the session cookies to be reset and new session cookies set. This and following tag data will be associated with the new session.
        - **99999999_clogin**: 'Session' cookie – stores a 'session ID'. Client-Managed implementations store separate "e=" and "i=" subkeys, replacing the 99999999_expires cookie used by 'Coremetrics Managed' implementations. Set under the 'Cookie Domain' for 'Client Managed' implementations, or the 'Data Collection Domain' for Coremetrics-Hosted implementations.
        - **99999999_expires**: 'Timeout' cookie timestamp value. Set only for 'Coremetrics Managed' 1st Party implementations under the 'Data Collection Domain'.
    - Utility Cookies: Only set for 'Client Managed' implementations and set under the specified 'Cookie Domain'. Expiration: at end of session.

- **cmTPSet**: Used to collect technical properties on 1st pageview of the session. This cookie is created only for implementations using eluminate.js v4.1.2 or later.

- **cmRS**: 'Resend' cookie – used to persist automatically collected 'Link Click' data to destination pages. Not set until the visitor clicks on an instrumented link.

- Persistent cookie (1)

  - Set under the specified 'Cookie Domain' for 'Client Managed' implementations, or the 'Data Collection Domain' for 'Coremetrics Managed' implementations.

    - **CoreID6**: Persistent cookie – stores a 'Visitor' ID value. Expiration: 20 years from date set.

8.7.1 Client Managed First Party Cookie Migration

Client Managed First Party implementations set Coremetrics cookies under the 'Cookie Domain' specified in parameter 4 of the cmSetClientID(…) function. In cases where visitors traverse multiple 2nd level domains as part of a normal site visit sending data to a single Coremetrics Client ID, it may be necessary to migrate the Coremetrics 'Visitor' and/or 'Session' cookies on the prior domain to the new 2nd level domain to maintain Visitor and Session continuity in reporting. This is necessary in order to insure complete data attribution in all reporting.

For this purpose Coremetrics provides the function 'cmSetupCookieMigration(…)'. Use of this function is recommended only in cases where pages served from the new 2nd level domain are typically visited after pages served by the original domain. This function call should be placed on all pages served from both the originating and destination domains, after the Coremetrics library src= include and cmSetClientID(…) call (if any), and prior to the cmSetProduction(); call (if any) and any tag function calls on the page.

Legacy Note: This function may not be available for implementations completed prior to April 2010. Please contact Coremetrics Support to upgrade.

Parameters

Parameter	Required	Description
JSFPmigration	Required	Always true
forceVisitorOverwrite	Required	true or false. This should be true unless otherwise advised by Coremetrics Support.

Parameter	Required	Description
domainWhitelist	Optional	Comma-delimited list of domains to which cookies should be transferred.
domainBlacklist	Optional	Comma-delimited list of domains to which cookies will not be transferred. Cookies will be migrated to all domains not in this list.
pathWhitelist	Optional	Comma-delimited list of URL paths to which cookies will be transferred. This is necessary when a 302 redirect is forwarding the visitor directly from the original domain to the new domain.
otherCookies	Optional	Comma-delimited list of other cookies to be migrated. (…,"cookie1,cookie2",…)
otherCookiesExpireTimes	Optional	Object Literal defining a list of optional expiration times for specified otherCookies. (…,{"cookie1":"1234567890",…},…)

One of the following three parameters must be specified: domainWhitelist, domainBlacklist or pathWhitelist. In cases where both a domain list and pathWhitelist is passed, pathWhitelist will be evaluated and domainWhitelist/domainBlacklist ignored.

- Example #1: domainBlacklist migration from domain 1 to domain 2 (no redirect)

Visitor browses the site and products under domain #1, 'thesite.com'. After carting an item and beginning checkout, the visitor is linked to a page served by new domain #2, 'checkout-store.com'. The Coremetrics Client Managed cookies should be migrated from 'thesite.com' to the new domain 'checkout-store.com' using the function 'cmSetupCookieMigration(…)' on all pages of domain #1 (thesite.com) and domain #2 (checkout-store.com).

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
<script type="text/javascript">
      cmSetupCookieMigration(true,true,".checkout-store.com");
</script>
</head>
```

```
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("Checkout: login", "checkout");
</script>
</body>
```

Note that pages served from the new domain, checkout-store.com, should include updated calls to cmSetClientID(…) setting Cookie Domain under the new site domain:

```
cmSetClientID("99999999",true,"data.coremetrics.com","checkout-store.com");
```

- Example #2: pathWhitelist migration for 'checkout.asp' after 302 redirect.

Visitor browses the site and products under domain #1, 'thesite.com'. After carting an item and beginning checkout, the visitor is redirected (302) to a page served by new domain #2, 'checkout-store.com'. The Coremetrics Client Managed cookies should be migrated from 'thesite.com' to the new domain 'checkout-store.com' using the function 'cmSetupCookieMigration(…)' on all pages of domains #1 (thesite.com) and #2. The pathWhitelist will contain "checkout.asp", uniquely identifying the URL of the first page served by the new domain after the 302 redirect.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
<script type="text/javascript">
      cmSetupCookieMigration(true,true,null,null,"checkout.asp");
</script>
</head>
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("Checkout: login", "checkout");
</script>
</body>
```

- Example #3: domainBlacklist migration + 2 other session cookies and 3 other persistent cookies with specified expiration periods in standard JavaScript millisecond date value.

```
<head>
<script type="text/javascript" src="//libs.coremetrics.com/eluminate.js"></script>
<script type="text/javascript">
      cmSetClientID("99999999",true,"data.coremetrics.com","thesite.com");
</script>
<script type="text/javascript">
      cmSetupCookieMigration(true, true, ".checkout-store.com", null, null,
"sessioncookie1,sessioncookie2,persistentcookie1,persistentcookie2,persistentcookie
3",{"persistentcookie1":1234567890,"persistentcookie2":987654321,"persistentcookie3
":2940792846});
</script>
```

```
</head>
<body>
...
<script type="text/javascript">
cmCreatePageviewTag("Checkout: login", "checkout");
</script>
</body>
```

8.8 *Impression Attribution*

The Coremetrics Marketing Impression Attribution is an optional module allowing tracking of 'impressions' from offsite marketing placements directing traffic back to a Coremetrics-tagged web site. Through sophisticated cookie mapping and attribution logic, it is possible to understand basic information like impressions, clicks and click-throughs in addition to advanced metrics like "attributed" sales, orders, sessions and events over defined attribution windows and selected credit logic. This module may require a tag library update. To determine if you need an update, please contact your assigned Implementation Engineer or Customer Support.

Using Impression Attribution, it is possible to answer questions such as:

- Am I underinvested in Display Advertising with a particular publisher or network?
- Do those who convert on my site ever view our ads? Where do they see them?
- Does someone who views our "High Interest Checking" ad banner on a particular site sign up for a new account on our site within 90 days?
- How often is a particular ad viewed across all sites? By how many unique viewers?
- Do those who watch our syndicated video or view our micro-sites ultimately convert on our site?
- What components of our widget are people viewing/utilizing? Does interest in a particular widget component signal a high likelihood for a downstream "Home Goods" purchase?

8.8.1 Independent Request

In order to limit the impact on performance/latency on the publishing site, independent requests are preferred. In the case of a micro-site, the request may simply be included directly into the page source. In the case of a display ad, the request would be called simultaneous with the ad request. In other words, the publishing site code would make a request for both the Coremetrics Marketing Impression Tag and the actual ad creative at the same time.

- Sample Independent Request:

  http://data.cmcore.com/imp?tid=17&ci=11111111&vn1=4.1.1&vn2=imp&ec=UTF-8&cm_mmc=Washington%20Post-_-Display%20Ads-_-Boat%20Insurance-_-Boat_vA_c432&cm_mmca1=300x250&cm_mmca2=german

8.8.2 Redirect Request

The 'redirect' request leverages a redirect through which the publishing site requests the Coremetrics Marketing Impression tag and it in turn redirects to the ad creative. The redirect URL (?ul=) must come before the tag parameters.

- Sample Redirect Request:

http://data.cmcore.com/imprd?ul=http://ad.doubleclick.com/ad/N733/B32049394.2%3Bsz%3D300x250%26lang%3Dgerman&tid=17&ci=11111111&vn1=4.1.1&vn2=imp&ec=UTF-8&cm_mmc=Washington%20Post-_-Display%20Ads-_-Boat%20Insurance-_-Boat_vA_c432&cm_mmca1=300x250&cm_mmca2=german

8.8.3 Marketing Impression Tag Specification

The Marketing impression tag is a light weight direct image request to Coremetrics that will set and/or collect a 3rd party cookie (CoreID6). Coremetrics records an "impression" for each tag received. Data.cmcore.com responds to requests with a transparent 1x1 pixel GIF image.

Parameters

Parameter	Required	Description
Protocol	Required	http:// or https://
Request Domain	Required	The Coremetrics domain receiving this request – always use "data.cmcore.com"
Requested File	Required	"/imp?" is used for an independent request. "/imprd?" is used for a redirect request.
Tag ID (tid=)	Required	"17" indicates a 'Marketing Impression' – always use this value.
&vn1=4.1.1 &vn2=imp &ec=UTF-8	Required	Always include these parameter values as demonstrated.
Marketing Program (cm_mmc=)	Required	Four-level definition of the Marketing Program using the standard "vendor-_-category-_-program-_-item" convention (see 4.1). This should match the associated marketing URL values.
Marketing Attributes (cm_mmca1= through cm_mmca15=)	Optional	Up to 15 "-_-" delimited 100-byte 'attribute' values. This should match the associated marketing URL values.

8.8.4 Marketing Program & Impression Correlation

The Marketing Program link URL and associated Marketing Impression requests for the same creative should contain the same Marketing Program (cm_mmc=) and Marketing Attribute (cm_mmca1-a15) parameter values.

- For the following display Ad example linking to your site's home page:

http://data.cmcore.com/imp?tid=17&ci=11111111&vn1=4.1.1&vn2=imp&ec=UTF-8&cm_mmc=**Washington%20Post-_-Display%20Ads-_-Boat%20Insurance-_-Boat_vA_c432**&cm_mmca1=**300x250**&cm_mmca2=**german**

- The target URL should contain:

http://www.yoursite.com?cm_mmc=**Washington%20Post-_-Display%20Ads-_-Boat%20Insurance-_-Boat_vA_c432**&cm_mmca1=**300x250**&cm_mmca2=**german**

Matching Marketing Program parameters and Marketing Attributes across the Marketing Impression and Marketing URL insures correct marketing reporting.

8.9 *Enterprise Product Report*

The Enterprise Products Report (EPR) provides access to a second Product / By Category report of product activity (sales, abandonment, items sold, productviews, etc). Each product in the EPR is associated with a single category through the creation of two required .csv files: 'Enterprise Product Category Mapping File' (EPCMF) and 'Enterprise Category Definition File' (ECDF). No new tagging is required. The new .csv files must be created and uploaded to Coremetrics through https://import.coremetrics.com. The EPR reports each product within the single associated product-category for that product, as differentiated from the 'Products / Product Categories' report which may report activity for the same product across multiple categories depending on product placements and visitor navigation to the various placements.

The ECDF and EPCMF can also be used to upload static product attribute data to Coremetrics for use with Intelligent Offer, even if the Enterprise Product Report is not activated for your account.

8.9.1 Enterprise Category Definition File (ECDF)

The 'Enterprise Category Definition File' must be created and uploaded for each Coremetrics Client ID to use the EPR. Similar to the standard 'Category Definition File' (CDF) described in section 8.1, the ECDF defines the hierarchical structure of categorized reporting to be displayed in the EPR.

- **ECDF Name**

"**ECDF_99999999_<clientfilename>.csv**" where 99999999 is the Coremetrics Client ID to receive the file. <clientfilename> can be specified for auditing purposes, such as a date or version value.

- **ECDF Format**

The ECDF definition and format is identical to that of the CDF, except the ECDF hierarchy defines a static Product hierarchy for reporting of each product in a single assigned branch of the hierarchy. The ECDF will not contain 'dynamically' assigned product placement categories such as "SEARCH", "CROSS-SELL", etc. Each Category ID can occur only once in the ECDF. See section 8.1.3 for information on use of quotes/doublequotes in the Category Name column value.

File Date	Client ID	Category ID	Category Name	Parent Category ID
YYYYMMDD	99999999	101GR	Mens	
YYYYMMDD	99999999	201GR	Womens	
YYYYMMDD	99999999	102GR	Mens Shirts	101GR
YYYYMMDD	99999999	103GR	Mens Pants	101GR
YYYYMMDD	99999999	202GR	Womens Shirts	201GR
YYYYMMDD	99999999	203GR	Womens Pants	201GR

- **ECDF Example Records**

YYYYMMDD,99999999,101GR,MENS,
YYYYMMDD,99999999,201GR,WOMENS,
YYYYMMDD,99999999,102GR,MENS SHIRTS,101
YYYYMMDD,99999999,103GR,MENS PANTS,101
YYYYMMDD,99999999,202GR,WOMENS SHIRTS,201
YYYYMMDD,99999999,203GR,WOMENS PANTS,201

8.9.2 Enterprise Product Category Mapping File (EPCMF)

The EPCMF file associates each unique Product ID with a single Category ID in the ECDF. The EPCMF will typically have a number of rows equal to the total unique Product ID values reported in Product View and Shop Action tags sent from the related web site(s). Additionally, this file provides for import of up to 15 static Product 'attributes' for each product. These 'attributes' then become available for display as data columns in the EPR. Note that these 15 'static' attributes are seperate from and in addition to the standard 50 product-related Explore 'attributes' collected through Product View and Shop Action tags. Maximum length for each 'static' attribute is 2000 characters.

- **EPCMF Name**

"**EPCMF_99999999_<clientfilename>.csv**" where 99999999 is the Coremetrics Client ID to receive the file. <clientfilename> can be specified for auditing purposes, such as a date or version value.

- **EPCMF Format**

In this example, static attribute 1 is specified. Up to 15 additional columns can be included for static attributes. See section 8.1.3 for information on use of quotes/doublequotes in the Product Name column value.

File Date	Client ID	Product ID	Product Name	Product Category ID	Static Attribute 1
YYYYMMDD	99999999	21344M	Long Sleeve Shirt	102GR	LEVIS
YYYYMMDD	99999999	5032M	Pleated Classic Fit Pants	103GR	DOCKERS
YYYYMMDD	99999999	1021W	Embroidered Logo Tee	202GR	ADIDAS

- **EPCMF Example Records**

YYYYMMDD,99999999,21344M,Long Sleeve Shirt,102GR,LEVIS
YYYYMMDD,99999999,5032M,Pleated Classic Fit Pants,103GR,DOCKERS
YYYYMMDD,99999999,1021W,Embroidered Logo Tee,202GR,ADIDAS

- **Notes**

    - The Product Category ID value in column 5 must match the Category ID of one record in the ECDF.
    - The Product ID value in column 3 must match the Product ID value sent with the Product View and Shop Action tags for the same Product.
    - Each row in the file must have the same number of columns: the place of each column for which a value is not sent must be identified through additional intervening or trailing commas, as necessary. For example, if a static attribute value #2 is provided in column 7 of record 1, but no static attribute #2 value is provided in record 2, the 2nd record must end with an additional comma. For example:

        - YYYYMMDD,99999999,21344M,Long Sleeve Shirt,102GR,LEVIS,**FITTED**
        - YYYYMMDD,99999999,5032M,Pleated Classic Fit Pants,103GR,DOCKERS**,,**

8.9.3 Uploading the ECDF and EPCMF

The ECDF and EPCMF should be uploaded to Coremetrics through the CDF import tool at https://import.coremetrics.com. This tool allows direct uploads of ECDF/EPCMF through the gui or creation of an ftp (port 22) or sftp (port 998) server account with Coremetrics (ftp.coremetrics.com) for scheduled upload of automatically generated files.

8.10 *Report and Tag Matrix*

An X indicates that the report is populated in whole or in part by the tag listed.

	Page View	Order	Tech Props	Link Click	Link Imp'n	Form Action	Product View	Shop Action	Registration	Element Tag	Event Tag
Top Line Metrics	X	X					X		X		X
Monitor	X	X					X	X	X	X	X
Visitor Purchase Funnel	X	X					X	X	X		
Session Purchase Funnel	X	X					X	X			
Session Event Funnel	X										X
Geography	X	X									
Demographics & System			X								
Marketing Channels		X									X
Marketing Programs		X									X
Marketing Zoom											X
Natural Search		X									X
Referring Sites		X									X
Multi-Source Marketing		X									X
Product Categories							X	X			
Product Zoom							X	X			
Top Browsed Items							X	X			
Top Abandoned Items							X	X			
Cross Sold Items								X			
Onsite Search	X	X						X			
Page Categories	X	X									X

	Page View	Order	Tech Props	Link Click	Link Imp'n	Form Action	Product View	Shop Action	Registration	Element Tag	Event Tag
Page Zoom	X	X									
Top Visited Pages	X										
Top Entry Pages	X										
Top Departure Pages	X										
Real Estate	X	X		X	X						X
Site Promotions	X	X		X	X						X
Elements										X	
Click Stream	X	X									
True Path	X	X						X			
Form Analysis						X					
Form Analysis Zoom						X					
LIVEview	X	X		X		X					X
Inbound Link Analysis	X	X									X
On-Site Link Analysis	X	X									
Report Segments	X	X	X	X					X		X
Segmentation - Overview	X	X	X						X		
Segmentation - Top Pages	X						X				
Segmentation - Top Items		X									
Segmentation - Email Extract									X		
Segmentation - Visitor	X	X					X		X		
Segmentation - Buyer		X					X	X	X		

8.11 *Patent Information*

Coremetrics products and services are licensed under the following Netratings patents: 5,675,510; 5,796,952; 6,115,680; 6,108,637; 6,138,155; 6,643,696 and 6,763,386.