Confidential

OMNITURE™

Engineering Services Application Assessment

Programming Exercise

Omniture Engineering Services

Objective

This exercise is to help us assess your level of understanding in PHP and problem solving. The following exercise is a small example of a potential custom project the Engineering Services may engage in. We are interested in how effective you are in answering the question, the steps you took to answer the question and any additional insight to help the customer gain more value from this process.

Setup

Attached is a simple tab separated file which contains what we call “hit level data”. A hit level record is a single “hit” from a visitor on the client’s site. Based on the client’s implementation, several variables can be set and sent to Omniture for deeper analysis.

Your exercise is to write a PHP application that is capable of reading this hit level data file and answer the client’s question. Review Appendix A for more details on how the hit level data file is formatted.

Development Exercise

The Client's Question

How much revenue is the client getting from external Search Engines, such as Google, Yahoo and MSN, and which keywords are performing the best based on revenue?

Development Requirements

1. Create a PHP application that executes from the command line
2. The PHP application needs to be broken up into a runtime script and at least 1 class
3. The runtime script needs to accept a single argument, which is the file that needs to be processed
4. The class can be whatever you want it to be but it needs to be used by the runtime script

Deliverable Requirements

The final output needs to be a tab delimited file with the following data points:

- Search Engine Domain (i.e. google.com)
- Search Keyword (i.e. "Laffy Taffy")
- Revenue (i.e. 12.95)

The final output also has the following requirements:

- A header row needs to be included. Use the above bulleted items for each column header, minus the example.
- Sorted by revenue, descending, so the client can easily review which keyword is performing the best.
- The output file should have the following naming convention: [Date]_SearchKeywordPerformance.tab
    - [Date] corresponds to the date the application executed for
    - The format should be YYYY-mm-dd (i.e. 2009-10-08)

Timeline

If you have any questions, you are encouraged to reach out to us for any clarity.

Verification

We will run the PHP application against the source file to verify that the requirements are met. You may want to consider compressing (i.e. Zip) the application to ensure it is delivered without spam filters catching it.

Other Considerations

Because our group deals with extremely large files, over 10 GB per file uncompressed. Give us your thoughts on how well this application will scale and if not, what improvements could be made to ensure it can scale to this file size.

Please consider this as a project you would develop for a client while you were working on the ES team. So take into consideration good coding practices.

Appendix A: Explanation of the Hit Level Data File

In order to achieve your objective, you will need to parse this file review the following columns:

- Hit_time_gmt
  - The time in GMT which the hit occurred
- Date_time
  - A friendly date time stamp, localized to the client's time zone
- User_agent
  - The user agent string from the browser
- IP
  - The IP Address of the visitor
- Event_list
  - A comma separated list of events that occur during the visit. Below is an explanation of the events.
  - Example format: "2,200,201,100"
    - 1: Purchase
    - 2: Product view
    - 10: Shopping Cart Open
    - 11: Shopping Cart Checkout
    - 12: Shopping Cart Add
    - 13: Shopping Cart Remove
    - 14: Shopping Cart View
    - 1[00-49]: instance of eVar1 – eVar 80 (these numbers are "0" based, i.e. 100 => eVar 1)
    - 2[00-79]: event1 – event 80 (these numbers are "0" based, i.e. 200 => event 1)
- Geo_City, Geo_Region, Geo_country
  - The geo location(s) of the visitor based on their IP address lookup
- Pagename
  - The friendly name of the page URL
- Page_url
  - The full URL of the page the visitor hit
  - Example: http://www.mysite.com/products/abc/
- Product_list
  - Any products the visitor is interacting with. For details on how the product_list is formatted, please review Appendix B
- Referrer
  - The referring URL to the page the visitor is currently reviewing
  - Example: http://search.yahoo.com/search?p=marketing&sm=Yahoo%21+Search

Appendix B: Explanation of the Product List variable

The products list is a comma (,) delimited list of products with a semi-colon (;) delimited list of attributes for each product

Example Format

[Category];{Product Name];[Number of Items];[Total Revenue];[Custom Event]|[Custom Event];[Merchandizing eVar],…

Real Example

"Computers;HP Pavillion;1;1000;200|201,Office Supplies;Red Folders;4;4.00;205|206|207"

Product Attribute Explanation

- Category: The category for the product (i.e. Shoes, Clothes)
- Product Name: Either the product ID or the product name
- Number of Items: The number of products
- Total Revenue: The price of the product. Revenue is only actualized when the purchase event is set in the events_list
- Custom Event: Events only applied to a specific product
- Merchandising eVar: eVars only applied to a specific product