COUNTER Code of Practice

Journals and Databases

Release 2

Appendix D

Guidelines for Implementation

Introduction

For ease of reference, the numbering used in this Appendix corresponds exactly to that of the Code of Practice itself; where appropriate the relevant section of the Code of Practice text is quoted.

5a: *'Only successful and valid requests should be counted. For webserver-logs successful requests are those with a specific return code. The standards for return codes are defined and maintained by NCSA.'*

Requirement for Implementation:
Return codes that indicate a successful or valid request are specified in agreed, international web standards and protocols. The relevant governing document for hypertext protocols is RFC2068, which contains definitions for each Return Code number. There are five categories of return code numbers:

1xx (Information): this category provides information on a request, and often indicates that the user has come upon an experimental application.
2xx (Success): reserved for successful responses. This category of code is not usually seen by the user, but their browser will receive them and will know that whatever request was sent by the browser was received, understood and accepted.
3xx (Redirection): indicates the need for further action by the user's browser. User action may not be necessary, as the browser may deal with it automatically.
4xx (Client Error): this category of code is the one most frequently seen by the user and indicates an error.
5xx (Server Error): indicates where the server knows it has made an error, or is not capable of answering the request.

Categories **2xx** and **3xx** are relevant to Section 5a of the COUNTER Code of Practice, which deems that **only the following specific return codes indicate a successful or valid request:**

200 (OK) The request was successful and information was returned. This is, by far, the most common return code on the web.

304 (Not modified) In order to save bandwidth a browser may make a conditional request for resources. The conditional request contains an 'If-Modified-Since' field and if the resource has not changed since that date the server will simply return the 304 code and the browser will use its cached copy of the resource.

Requests that result in any other return codes within the 2xx and 3xx categories must not be counted. This exclusion covers:

206 (Partial content) This indicates that the server has only filled part of a specific type of request.
301 (Moved permanently): The addressed resource has moved, and all future requests for that resource should be made to the new URL. Transfer to the new location may be automatic or may require manual intervention by the user. Either way, a successful request to the new location will generate a 200 return code.
302 (Moved temporarily) This indicates that the content has moved while the page requested still has the same URL. The page is, therefore, not retrieved and must not be counted.
303 (See other) The response to the browser's request can be found elsewhere. Automatic redirection may take place to the new location.

Full information on the five categories of http return codes and their definitions may be found at: http://www.w3.org/Protocols/rfc2068/rfc2068 goto: Chapter 10 (pp 53-64): Status Code Definitions. More summarised information may be found at: http://www.cknow.com/ckinfo/def_h/httpreturncodes.shtml .

5e. Guidelines for processing and filtering the raw usage data

The filtering of the 'raw' usage data needs to go through a number of consecutive steps in order to meet the COUNTER requirements.

Step1: Sorting the data file.

The file to be used for reporting should be sorted chronologically by user. The following options for a user exist:

1. Where only the IP address of a user is logged that IP should be taken as the field to sort by.
2. When a session-cookie is implemented and logged, the session-cookie should be used to sort by.
3. When user-cookies are available and logged, the user-cookie should be used to sort by.
4. When the username of a registered user is logged, this username should be used to sort by.

Step 2: Remove all records with a return code other then '200' and '304'

Step 3: Run the 'double-click-removal' script

The following example illustrates how this script should work:

A user requests the HTML version of one and the same article four times within the following time intervals:

Request 1: 9:51:10

Request 2: 9:51:19

Request 3: 9:51:32

Request 4: 9:51:41

Applying the double-click filter to the above example has the following results: comparing Requests 1 and 2 removes Request 1 and retains Request 2; next, comparing Request 2 with Request 3, retains both Request 2 and Request 3 as more than 10 seconds have elapsed between these two requests; next, comparing Request 3 with Request 4 removes Request 3 and retains Request 4, as less than 10 seconds have elapsed between Requests 3 and 4. Thus, applying the double-click filter to the above example results in two Successful Requests being recorded.