Paper 079-2012

A Perfect Case of Capturing Data from Related Web Pages

Jinson J. Erinjeri, D.K. Shifflet and Associates Ltd., McLean, VA

ABSTRACT

The World Wide Web generates information at a very fast pace and it can become equally important to capture the data associated with the information at that pace. This paper deals with an application that captures the data from related web pages and converts it into a SAS® data set. This application converts the source code of a web page in text format to a SAS data set using Base SAS® and the same is applied across related web pages using SAS macros

INTRODUCTION

Manual entry of data from web pages and the copy-paste options are laborious as well as time consuming. The best approach would be to capture data from all the related web pages with the least human involvement in one stroke. This will not only save time but avoid errors while disseminating information. The case application presented here is one that was developed while trying to update our database of hotels.

PROBLEM: To obtain all the hotels under Best Western portfolio on their website by states and classify them as regular/ plus /premier into a SAS data set.

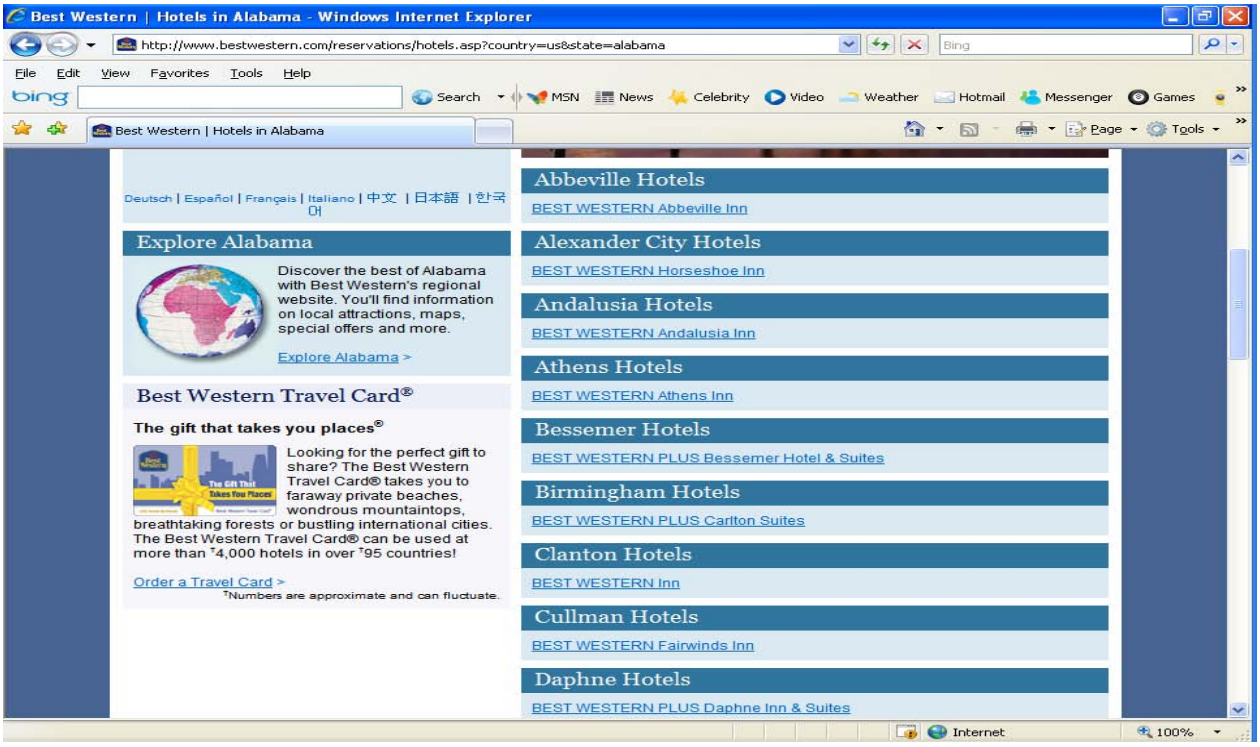

Figure 1. Screenshot of Webpage http://www.bestwestern.com/reservations/hotels.asp?country=us&state=alabama

The first task is to obtain the name of the city and the Best Western hotels located in that city from the webpage for a given state. For example, the list of Best Western hotels for the state of Alabama is shown in Figure 1 and the corresponding webpage is http://www.bestwestern.com/reservations/hotels.asp?country=us&state=alabama. The text of this webpage can be viewed from the web browser and is presented in Figure 2. The highlighted text in Figure 2 is the kind of data we need to extract. The second task is to obtain similar data for all the states in the United States. For example, the associated web page for the state of Virginia is http://www.bestwestern.com/reservations/hotels.asp?country=us&state=virginia. Note that the only difference between the two web pages is the name of the state. This is an ideal situation to apply SAS Macros across all states. The final task of classifying the hotels into regular/plus/premier can be achieved by manipulating the extracted data.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Best Western | Hotels in Alabama</title>
<link rel="shortcut icon" href="/images/favicon.ico">
<script language="JavaScript" type="text/javascript" src="/scripts/global.js"></script>
<script language="JavaScript" type="text/javascript" src="/scripts/globalfunc.js"></script>
……
<meta name="keywords" content="Alabama hotels, Best Western, family friendly hotels in Alabama, affordable hotel in Alabama, hotel locations in Alabama, hotel accommodations in Alabama, find a hotel in Alabama">
</head>
<body>
……
……..
                                                    <img src="/images/bestwestern/states/al.jpg" width="451" height="204" alt="Alabama hotel listings. Find an affordable hotel with quality hotel accommodations." border="0">
                                                    <table cellpadding="0" cellspacing="0" border="0" width="100%" style="margin-top:8px;margin-bottom:8px;">
                                                        <tr><td class="hdrWhiteOnMidBlue"><h2>Abbeville Hotels</h2></td></tr>
                                                    <tr><td class="containerLightBlue"><a target="_top" href="http://book.bestwestern.com/bestwestern/productInfo.do?iata=&promoCode=&corpID=&propertyCode=01065" >BEST WESTERN Abbeville Inn</a></td></tr>
</table>
                                                    <table cellpadding="0" cellspacing="0" border="0" width="100%" style="margin-top:8px;margin-bottom:8px;">
                                                        <tr><td class="hdrWhiteOnMidBlue"><h2>Alexander City Hotels</h2></td></tr>
                                                    <tr><td class="containerLightBlue"><a target="_top" href="http://book.bestwestern.com/bestwestern/productInfo.do?iata=&promoCode=&corpID=&propertyCode=01087" >BEST WESTERN Horseshoe Inn</a></td></tr>
</table>…….
                                                    <table cellpadding="0" cellspacing="0" border="0" width="100%" style="margin-top:8px;margin-bottom:8px;">
                                                        <tr><td class="hdrWhiteOnMidBlue"><h2>Birmingham Hotels</h2></td></tr>

                                                    <tr><td class="containerLightBlue"><a target="_top" href="http://book.bestwestern.com/bestwestern/productInfo.do?iata=&promoCode=&corpID=&propertyCode=01096" >BEST WESTERN PLUS Carlton Suites</a></td></tr>
……..
……
```

Figure 2. Partial Text of the webpage
http://www.bestwestern.com/reservations/hotels.asp?country=us&state=alabama

It is important to note that most of the text in Figure 2 needs to be filtered out and only a small portion is required to be captured for the problem in hand.

SOLUTION:

The main steps involved in capturing the Best Western hotels data for the state of Alabama from its webpage is presented in this section. The SAS macro program for obtaining the data for all the states in the US is presented in the Appendix. The comments in the code provide the details of the program.

Step 1: SAS provides a URL access method for accessing websites via the FILENAME statement. This statement performs the basic HTTP requests and responds to the website stated in Figure 1. In the code below web_loc is the fileref and "&pf" resolves to http://www.bestwestern.com/reservations/hotels.asp?country=us&state=alabama.

```
filename web_loc url "&pf." debug;

data source;
      format webpage $1000.;
      infile web_loc lrecl=32767 delimiter=">";
      input webpage $ @@;
```

```
run;
```

The debug option in the FILENAME statement tells SAS to display the HTTP headers in the SAS log. The LRECL option in the INFILE statement overrides the default record length line of 256 to 32767 to avoid truncation. In Figure 2, the highlighted text is of interest and it is always preceded by the symbol ">" and therefore the DELIMITER option was set to ">" to parse out the needed text. The @@ option places each value in the next observation instead of placing it in the next variable. The partial output of Step1 is presented in Figure 3.

Webpage
<td class="containerLightBlue"
<a target="_top" href="http://book.bestwestern.com/bestwestern/productInfo.do?iata=&promoCode=&corpID=&propertyCode=01096"
BEST WESTERN PLUS Carlton Suites</a
</td
</tr
</table
<table cellpadding="0" cellspacing="0" border="0" width="100%" style="margin-top:8px;margin-bottom:8px;"

Figure 3. Partial Output of Step 1.

Step 2: Using various SAS functions, we can delete the unwanted observations. The SCAN function is used to obtain the first word and the ANYPUNCT function returns the position of first occurrence of punctuation from the list of ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~. It returns a value of 0 if none is present and this is used to delete the observations with any punctuation.

```
data sources;
      set source;
      k2=scan(webpage,1,1);
      any=anypunct(k2);
run;
```

Further deletions are carried out using the combination of ANYALPHA, ANYDIGIT and ANYPUNCT functions to delete observations with missing data. The VERIFY function is used to return the position where the string mismatch occurs which is named f1 in the code. Using the SUBSTR function we extract the last two characters and retain all the observations ending with '/a' or 'h2' and with f1>=13.

```
data sourcem;
      set sources;
      where any=0;
      vlamalpha=anyalpha(webpage);
      vlamdigit=anydigit(webpage);
      vlampunct=anypunct(webpage);
      if vlamalpha=0 and vlamdigit=0 and vlampunct=0 then delete;
      f1=verify(webpage,"BEST WESTERN");
      last_two = substr(webpage,LENGTH(webpage)-1,2);
      if f1>=13 and last_two='/a' or last_two='h2';
run;
```

webpage	k2	any	f1	vlamalpha	vlamdigit	Vlampunct	last_two
Abbeville Hotels</h2	Abbeville	0	1	1	20	17	h2
BEST WESTERN Abbeville Inn</a	BEST	0	14	1	0	27	/a
Alexander City Hotels</h2	Alexander	0	1	1	25	22	h2
BEST WESTERN Horseshoe Inn</a	BEST	0	14	1	0	27	/a
Andalusia Hotels</h2	Andalusia	0	1	1	20	17	h2
BEST WESTERN Andalusia Inn</a	BEST	0	14	1	0	27	/a
Athens Hotels</h2	Athens	0	1	1	17	14	h2
BEST WESTERN Athens Inn</a	BEST	0	14	1	0	24	/a
Bessemer Hotels</h2	Bessemer	0	2	1	19	16	h2
BEST WESTERN PLUS Bessemer Hotel & Suites</a	BEST	0	14	1	0	34	/a

Figure 4. Partial Output of Step 2.

Step 3: The index function is used to locate the position where "<" occurs and this in turn is used in the SUBSTR function to extract the needed characters. Using the scan function we can classify the hotels into regular/plus/premier. The partial output is shown in Figure 5.

```
data sourcef;
      set sourcem(keep=webpage);
      any1=index(webpage,'<');
      all_hotels=substr(webpage,1,any1-1);
run;

data sourceff;
      set sourcef;
      if scan(all_hotels,3)="PREMIER" then   desired='PREMIER';
      else if scan(all_hotels,3)="PLUS" then desired='PLUS';
      else                                   desired='REGULAR';
      state="alabama";
      keep state all_hotels desired;
run;
```

all_hotels	desired	state
Abbeville Hotels	REGULAR	alabama
BEST WESTERN Abbeville Inn	REGULAR	alabama
Alexander City Hotels	REGULAR	alabama
BEST WESTERN Horseshoe Inn	REGULAR	alabama
Andalusia Hotels	REGULAR	alabama
BEST WESTERN Andalusia Inn	REGULAR	alabama
Athens Hotels	REGULAR	alabama
BEST WESTERN Athens Inn	REGULAR	alabama
Bessemer Hotels	REGULAR	alabama
BEST WESTERN PLUS Bessemer Hotel & Suites	PLUS	alabama
Birmingham Hotels	REGULAR	alabama
BEST WESTERN PLUS Carlton Suites	PLUS	alabama
Clanton Hotels	REGULAR	alabama

Figure 5. Partial Output of Step 3.

CONCLUSION

Using SAS Base and SAS Macros, one can extract data from related web pages efficiently as illustrated with a real time example in this paper.

REFERENCES

1. http://support.sas.com/publishing/pubcat/chaps/59343.pdf

2. Helf, G., Hitachi Global Storage Technologies, San Jose, CA. "Extreme Web Access: What to DO When FILENAME URL is Not Enough" Proceedings of the SAS Users Group International 30. Cary, NC: SAS Institute Inc. Available at: http://www2.sas.com/proceedings/sugi30/100-30.pdf

ACKNOWLEDGMENTS

The author would like to thank Nandini Nadkarni and Susan McElheny for their valuable input while reviewing this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jinson J. Erinjeri
D.K. Shifflet and Associates Ltd.
1750 Old Meadow Rd., Suite, 620
Mclean, VA 22102
Work Phone: 703-536-0924
Fax: 703-536-0580
E-mail: jerinjeri@dksa.com
Web: www.dksa.com

APPENDIX

```
options mlogic mprint symbolgen;

%let allstate=alabama alaska arizona arkansas california colorado connecticut
delaware florida georgia hawaii idaho illinois indiana iowa kansas kentucky
louisiana maine maryland massachusetts michigan minnesota mississippi missouri
montana nebraska nevada new-hampshire new-jersey new-mexico new-york north-
carolina north-dakota ohio oklahoma oregon pennsylvania rhode-island south-
carolina    south-dakota tennessee texas utah vermont virginia washington
west-virginia wisconsin wyoming;
%let
p1=%nrstr(http://www.bestwestern.com/reservations/hotels.asp?country=us&state=
);

%macro get_all_hotels;

%do i=1 %to 50;
%let k=%scan(&allstate.,&i.,' ');
%put &k.;
%let pf=&p1.&k.;
%put &pf.;

      filename web_loc url "&pf." debug;

      data source&i.;
            format webpage $1000.;
            infile web_loc lrecl=32767 delimiter=">";
            input webpage $ @@;
      run;

      data sources&i.;
            set source&i.;
            k2=scan(webpage,1,1);/*retrieves the first word from variable
webpage so that we can capture valid hotels/city and exclude the special
characters*/
            any=anypunct(k2);/*returns the position of the first occurrence of
punctuation from the list of punctuations*/
      run;

      data sourcem&i.;
            set sources&i.;
            where any=0;
            vlamalpha=anyalpha(webpage);
            vlamdigit=anydigit(webpage);
            vlampunct=anypunct(webpage);
            if vlamalpha=0 and vlamdigit=0 and vlampunct=0 then delete;
            f1=verify(webpage,"BEST WESTERN"); /*verify returns the position
from where the mismatch occurs*/
            last_two = SUBSTR(webpage,LENGTH(webpage)-1,2);/* obtains the last
two characters*/
            if f1>=13 and last_two='/a' or last_two='h2';
      run;

      data sourcef&i.;
            set sourcem&i.(keep=webpage);
            any1=index(webpage,'<');/*to obtain the position of where '<'
occurs */
```

```
            all_hotels=substr(webpage,1,any1-1);/*to delete the characters '<'
and right of it*/
      run;

      data sourceff&i.;
            set sourcef&i.;
            if scan(all_hotels,3)="PREMIER" then   desired='PREMIER';
            else if scan(all_hotels,3)="PLUS" then desired='PLUS';
            else                                   desired='REGULAR';
            state="&k";
            keep state all_hotels desired;
      run;

      %end;

      data complete;
            length state $17;
            set
                  %do p=1 %to 50;
                  sourceff&p.
                  %end;
            ;
      run;

      ods html file="BW_brands.xls" rs=none style=minimal;
            proc print data=complete;
            run;
      ods html close;

%mend;
%get_all_hotels;
```