

Speed it Up: Using SAS® to Automate Initial Discovery Practices

Mariya Karimova¹ and Christine John¹
¹AdvanceMed Corporation, an NCI Company Nashville, TN

Abstract

Healthcare investigations frequently begin with a tip containing very little provider information. This presentation attempts to use SAS® to automate the initial discovery process, turning a name into a full overview of the provider in order to identify unfounded complaints and accelerate the discovery of true issues. Multiple data sources are combined, which oftentimes require fuzzy matching to resolve conflicting identifiers. The program utilizes INFILE URL and SAS® 9.2 text functions to obtain meaningful information from various websites. It further utilizes SAS ODS and SAS/GRAPH® to create a single standard PDF report; which provides a visualization of provider billing patterns, summarizes their affiliations, and embeds hyperlinks to original web-based resources.

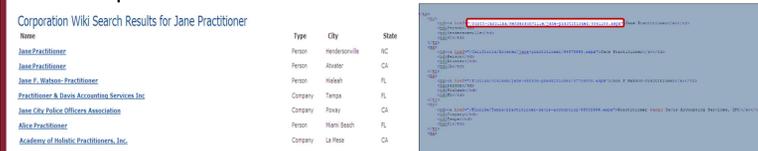
Web-Crawling

The example shown here is from Corporation Wiki. This exercise is purely academic in nature and is designed to show the capabilities of SAS. A programmer should consider copyrights and other legal implication before utilizing any web crawling techniques for a commercial use.

Identifying the Appropriate URL

Most URLs are easily parsable. Corporation Wiki search page has URL in a form <http://www.corporationwiki.com/search/results?term=firstname+lastname>

The figures below show the correlation between the search page the underlying HTML code and SAS code that can be used to return all potential URL results



```
FILENAME SEARCH_URL "&SLINK";  
DATA PROFILE_&NPI_WIKI2;  
INFILE SEARCH_LRECL=32767 ;  
LENGTH LINK3 $270;  
INPUT @ &LINK1.;  
LINK3=CAT('http://www.corporationwiki.com', '/',  
COMPRESS(&LINK1., ' '), SCAN(LINK1, 1, '>'));  
RUN;
```

SLINK in line 1 refers to a macro variable containing the concatenation of the base web address and known name.

Narrowing the Results

Decision rules are necessary to reduce multiple results down to one.

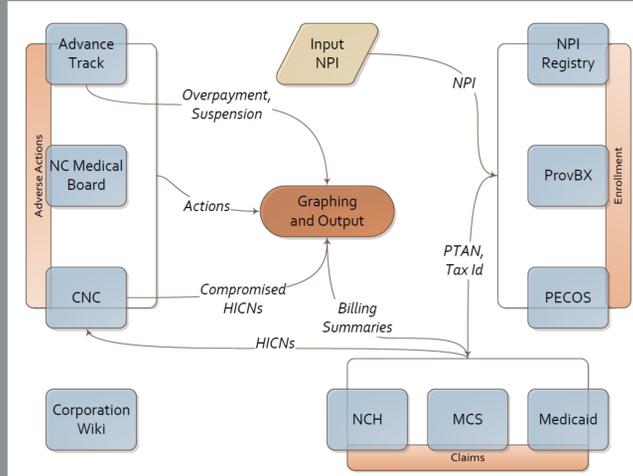
Consider: location, medical field affiliation, and known associates

```
%MACRO MULTIPLE;  
I = 1 %TO &N_OBS. ;  
PROC SQL NOPRINT;  
SELECT LINK3 INTO: LINK3 FROM  
WORK.PROFILE_&NPI_WIKI  
WHERE N_OBS=%EVAL(&I.) ;  
QUIT;RUN; %END;  
%MEND MULTIPLE;
```

Generalized Web-Crawling Algorithm

- 1) Study the structure of the websites HTML
- 2) Infile the search page using the search phrase (INFILE URL)
- 3) Use INPUT@ to identify the string containing the URL of the desired webpage
- 4) Use text functions to clean it and do the INFILE again
- 5) Write a macro to repeat step 4 until each possible webpage has been checked
- 6) Apply decision rules to identify matching page
- 7) Repeat steps 1-3 to get the information of interest

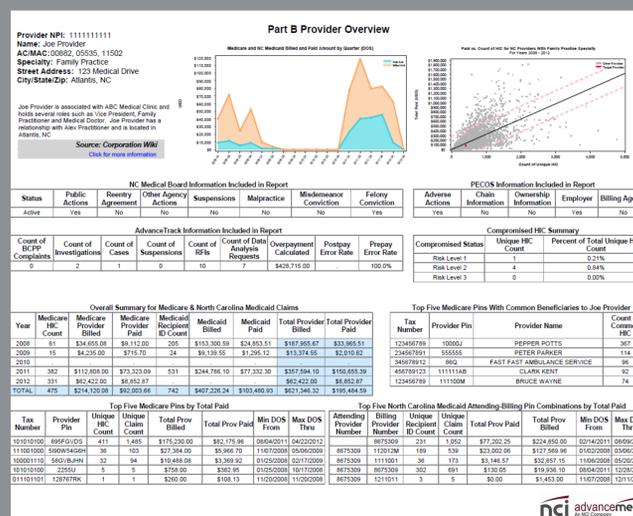
Discovery Process in Need of Automation



The figure above illustrates the process involved during a comprehensive provider discovery. Our goal was to automate this process using SAS® and return a concise and user-friendly report.

The majority of the automation process was performed using basic querying capabilities of SAS DATA steps and PROC SQL. This presentation focuses on the programming needed for the web crawling as well as the PDF output.

Desired Report for End User



The figure above is a representation of the end product. All actual information included in it (provider name, actions, billing summaries, etc) are fictional.

Output

Exporting directly to a PDF using SAS ODS PDF enables us to rapidly produce structured reports that require no additional "cleaning" after being exported and can contain any necessary combination of tables, graphs, and hyperlinks.

Inline Formatting – Not Just for Text, but Logos Too

Defining an escape character allows for inline formatting of the font size, font type, justification, color, etc.

```
Sample Code  
ODS ESCAPECHAR='^';  
...  
ODS PDF TEXT="^S={JUST=CENTER  
FONT_SIZE=12PT}Part B Provider Overview";
```

You can use it to add an image file anywhere you can put ODS text

```
Sample Code  
ODS PDF TEXT="^S={JUST=LEFT FONT_SIZE=13PT  
PREIMAGE='C:\LOGO.bmp?WIDTH=1.5IN&HEIGHT=0.5IN'}";
```

Compute Blocks

Individual tables were created using PROC REPORT. Special row and column formatting, such as the shading of specific columns or spanning table headers, was achieved through the use of COMPUTE blocks.

Compute blocks were also used to embed hyperlinks into the PDF

```
Sample Code  
PROC REPORT  
...  
DEFINE LINK_VAR / DISPLAY;  
COMPUTE GOOGLE;  
URLSTRING = http://www.google.com/;  
CALL DEFINE (_COL_, "URL", URLSTRING);  
ENDCOMP;  
...  
RUN;
```

ODS Regions

To output multiple charts and graphs on the same page, we used ODS REGION. This requires that prior to each procedure, you define the region on the page where that procedure should be printed.

```
Sample Code  
/*OUTPUT AREA LINE GRAPH OF BILLED/PAID*/  
ODS REGION  
X=2.75 IN Y=0.35 IN  
WIDTH=4.125 IN HEIGHT=2.125 IN;  
PROC GREPLAY IGOUT=WORK.BILLING_GRAPH NOFS;  
TC SASHELP.TEMPLT;  
TEMPLATE=WHOLE;  
TPLAY 1:PAID;  
QUIT;
```



Useful Functions

COMPRESS – removes specified characters from the original string

SCAN – returns n-th word from a character string

TRANWRD – replaces all occurrence of a substring in a character string

TRANSLATE - replaces specific characters in a character string.

TRIM - removes trailing blanks from a character string

STRIP - returns a character string with all leading and trailing blanks removed

FINDW - returns the character position of a word in a string, or returns the number of the word in a string

Lessons Learned

Writing a SAS script for other users with varying levels of SAS experience presents its own unique set of challenges. Detailed below are just a few of the best-practices that were identified during the development of this report.

Not All Users Regularly Clear Their Libraries

It became common for an analyst to run several reports in a single day, which was rapidly filling users' temporary libraries. The following adjustments became necessary additions to our script.

PROC DATASETS	Added to delete all intermediate tables
<i>DM 'ODSRESULTS; CLEAR;';</i>	Added to clear all the user's output window
PROC GREPLAY	Added to delete all graphs

User-Defined Macro Variables Run Amuck

There are several occasions throughout our program where information is identified from the results of a query and is assigned to a macro variable. In cases where a query would return no results, nothing is written to the macro variable. If a previous profile had been run which assigned a result to this macro variable, then it was not being overwritten, and would appear in parts of the second profile.

%SYMDEL macro	Added to delete all macros defined during the program
SQLQBS	Allows to control for number of observations returned by the last PROC SQL statement

Converting from SAS® 9.2 to SAS® 9.3

SAS 9.3 understands the specified location of the ODS REGIONS differently than SAS 9.2.

This prompted an immediate revision of the output section of the script, where each of the locations of the individual regions had to be re-defined.

About NCI

NCI's capabilities are centered on helping our customers overcome challenges to carry out their critical missions and meet their objectives. Our goal is to identify services and solutions that enable our customers to fulfill essential mission requirements at costs that are within budgetary expectations. We help our customers reduce costs and achieve new levels of performance by utilizing cutting-edge technology, such as those associated with IT consolidation and modernization, cybersecurity, information assurance, and cloud computing.

AdvanceMed, an NCI business unit, helps U.S. Federal agencies reduce improper payments by fighting fraud, waste, and abuse in entitlement programs. AdvanceMed utilizes analytic techniques and automated tools to stop fraudulent activity before it gets to the entitlement system and to aggressively identify fraudsters who have already made it into the system.

Contact the Authors

Mariya Karimova	email: karimovm@admedcorp.com
Christine John	email: johnc@admedcorp.com