

Use Base SAS FILENAME URL to Build Surveillance and Monitoring System for New Clinical Trial Registration

Xin Wei, James Cai, Jim Rosinski
Pharma Research Scientific Informatics
Hoffmann-La Roche, Nutley, NJ

ABSTRACT

Clinical trial is an irreplaceable approach to test the efficacy and safety of new drugs in human subjects. In recent years, the number of clinical trials sponsored by pharmaceutical industry, federal government or non-profit organizations has been rapidly increasing to develop new therapeutics for the unmet medical needs. Currently, Clinicaltrial.gov, the central repository for clinical trial registration developed by the national institute of health (NIH) and Food and Drug Administration (FDA), holds 80, 268 trials conducted in many nations and this database is expanding on the daily basis. To obtain the latest information on trials of interest require constantly checking the website, which may be time and labor consuming. It would be beneficial to patients, regulators and trial sponsors to have an automated data pipeline that extracts the latest data from registered trials by dynamically fetching the contents of on-line information in a pre-scheduled time frame. Here we demonstrate how to use base SAS® filename URL method to mine the data from Clinicaltrials.gov on real time. In addition, we also discuss how to launch a batch SAS task in a pre-define schedule and set up an email alert system to deliver the new findings to customer in a timely fashion. Throughout this paper, we demonstrate these utilities and functionalities with an example of reporting a new trial conducted in China sponsored by a big pharmaceutical company.

INTRODUCTION

The U.S. National Institutes of Health (NIH) and the Food and Drug Administration have developed www.Clinicaltrial.gov in light of passage of FDA Modernization Act in November 1997. The Food and Drug Administration Amendments Act of 2007 (FDAAA or US Public Law 110-85) was passed on September 27, 2007. The law requires mandatory registration and results reporting for certain clinical trials of drugs, biologics, and devices. These legislative mandates and IT infrastructure development make www.clinicaltrial.gov a reliable and comprehensive repository of clinical trial registry. This database serves as a great resource for the health care providers who are looking for the new hopes for patients suffering from terminal diseases and for the trial sponsors who conduct analysis of competitive intelligence. To meet these needs, the website does a good job allowing users to search for the latest trial information by a variety of parameters including disease type, sponsor, registered date, phase and locations etc. In this paper, we described a framework of using SAS URL access of filename statement as web browser to automatically retrieve data from the latest on-line trial information based on user-defined criteria. This process can be configured in bath mode so that the tasks are launched on timely schedule. The newly pulled-out information are either directly delivered to the end user via email alert or transferred to data warehouse for the future analysis. For the illustrative purpose, we use an example in which the user receives an email alert whenever the pharmaceutical company of user's interest registers a clinical trial in China targeting live carcinoma.

WORKFLOW

STEP 1: Fetch the data for all the trials that are conducted in China as “baseline” database

The idea of identifying the newly registered trial is to first establish a “baseline” dataset which contains all the data for the time being. All the upcoming ‘new’ trials that fit your search criteria will be identified by comparing the most updated dataset with the baseline dataset for historical records.

- 1.1 Determine the URL for the search of interest. First of all, search all the registered trials conducted in China in www.clinicaltrial.gov by clicking list studies by locations -> East Asia ->China. There are total 1452 studies currently registered in the data base as shown in Figure 1. Copy the URL for this search result as the starting point for the data retrieval by SAS FILENAME statement (URL access).

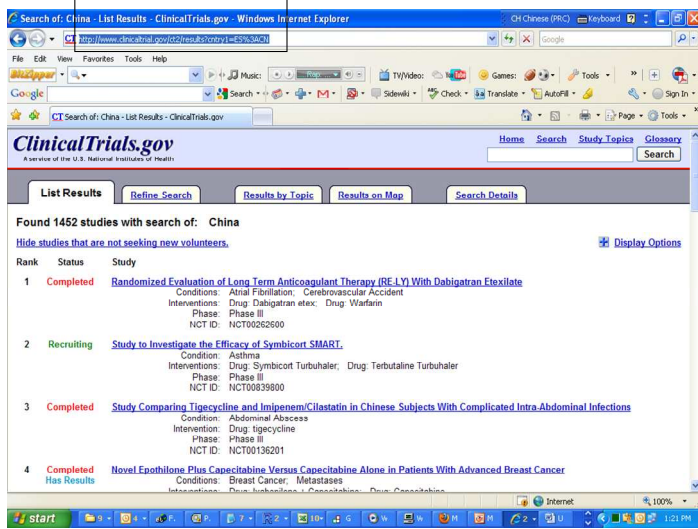


Figure 1

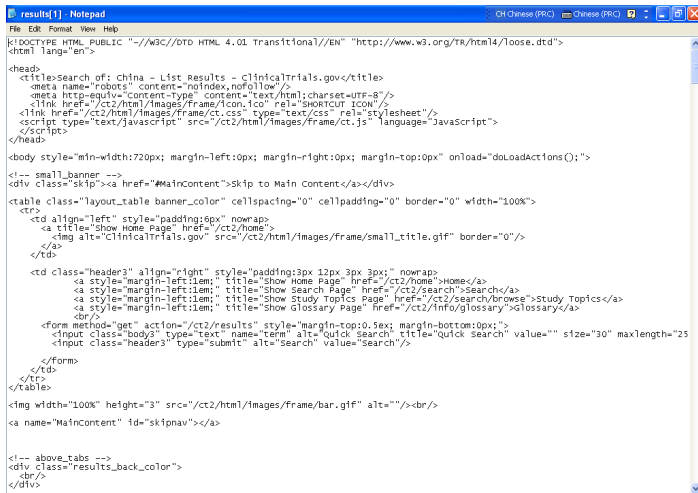


Figure 2

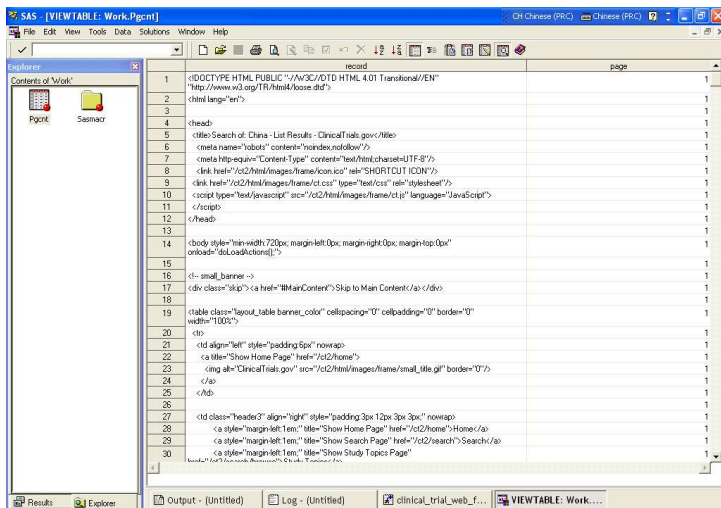


Figure 3

	study	url	id
1	Randomized Evaluation of Long Term Anticoagulant Therapy (RE-LY) With Dabigatran Etesate	/ct2/show/NCT00262007?cntry1=ES%3ACN&a	NCT00262000
2	Study to Investigate the Efficacy of Symbicort SMART.	/ct2/show/NCT00839800?cntry1=ES%3ACN&a	NCT00839800
3	Study Comparing Tigecycline and Impenem/Cilastatin in Chinese Subjects With Complicated Intra-Abdominal Infections	/ct2/show/NCT00136201?cntry1=ES%3ACN&a	NCT00136201
4	Novel Etoposide Plus Capecitabine Versus Capecitabine Alone in Patients With Advanced Breast Cancer	/ct2/show/NCT00080301?cntry1=ES%3ACN&a	NCT00080301
5	Efficacy and Safety Study of Apixaban for the Treatment of Deep Vein Thrombosis or Pulmonary Embolism	/ct2/show/NCT00643201?cntry1=ES%3ACN&a	NCT00643201
6	LOGIC - Lapatinib Optimization Study in ErbB2 (HER2) Positive Gastric Cancer: A Phase III Global, Blinded Study Designed to Evaluate Clinical Endpoints and Safety of Chemotherapy Plus Lapatinib	/ct2/show/NCT00680901?cntry1=ES%3ACN&a	NCT00680901
7	Nexava-1 aceva Combination Therapy for First Line Treatment of Patients Diagnosed With Hepatocellular Carcinoma	/ct2/show/NCT00901901?cntry1=ES%3ACN&a	NCT00901901
8	Study Of Celecoxib Or Diclofenac And Omeprazole For Gastrointestinal (GI) Safety in High GI Risk Patients With Arthritis	/ct2/show/NCT00141102?cntry1=ES%3ACN&a	NCT00141102
9	PRELUDE Study to Investigate the Prevention of Relapse in Lymphoma Using Daily Enasitumab	http://www.clinicaltrials.gov/ct2/results?cntry1=ES%3ACN&pg=1	NCT00332202
10	A Study to Assess the Safety and Efficacy of Adalimumab Administered as Subcutaneous Injections in Adult Chinese Rheumatoid Arthritis Subjects Treated With Methotrexate	/ct2/show/NCT00538902?cntry1=ES%3ACN&a	NCT00538902
11	A Study of Calcitriol (Mycophenolate Methyl) Combined With Tacrolimus and Corticosteroids in Kidney Transplant Patients.	/ct2/show/NCT00758602?cntry1=ES%3ACN&a	NCT00758602
12	The Efficacy of Human Acellular Dermal Matrix in the Treatment of Anal Fissure	/ct2/show/NCT00951002?cntry1=ES%3ACN&a	NCT00951002
13	Effectiveness and Safety of Ramipril Alone Compared With Telmisartan Alone and in Combination With Ramipril in Patients at High Risk for Cardiovascular Events: Patients Intolerant to Ramipril View Err	/ct2/show/NCT00153101?cntry1=ES%3ACN&a	NCT00153101
14	A Local Register Study For Major Depression Of Paroxetine Controlled Release	/ct2/show/NCT00368303?cntry1=ES%3ACN&a	NCT00368303
15	A Study to Compare Efficacy and Safety of Mycamine#174 and Inocazole for Preventing Fungal Infections	/ct2/show/NCT00794703?cntry1=ES%3ACN&a	NCT00794703
16	The Stabilization of Atherosclerotic Plaque by Initiation of	/ct2/show/NCT00799903?cntry1=ES%3ACN&a	NCT00799903

Figure 4

1.2 Determine the number of result pages that need to be converted to SAS dataset. SAS implements URL method of filename statement that can retrieve data from internet. This method works well with relatively simple web pages that open to public and do not require user login for the access to proprietary information. The following SAS code reads the text contents shown in figure 2 into a SAS data set.

```
filename fetch url
    'http://www.clinicaltrials.gov/ct2/results?cntry1=ES%3ACN&pg=1'
    debug lrecl=8192;

data pgcnt;
    infile fetch length=len;
    input record $varying8192. len;
    page=1;
run;
```

Besides the web page (Figure 1), the corresponding HTML codes and the textual contents in SAS data set are shown in Figure2 and Figure 3, respectively. The relevant information can be located and parsed out by SAS data step function such as index(), scan() and tranwrd(). For example, under the current setting, each search page only has 20 results. We need to determine how many page of search results need to be processed. The following SAS code achieves that:

```
data _null_;
    set pgcnt;
    if index(lowercase(record), 'found') and
        index(lowercase(record), 'studies with search of')
    then do;
        record=tranwrd(record, 'Found', '#');
        record=tranwrd(record, 'studies', '#');
        rd=input(scan(record, 2, '#'), best.);
        call symput('pg', ceil(rd/20));
        stop;
    end;
run;
%put %>>>> &pg;
```

In this code, the total number of hits “1452” is located by its co-occurrence with the text string “studies with search of” by index() function. The two words separated by “1452” are then replaced with “#” so that the number “1452” can be parsed out by scan(). Finally, the total number of pages that need to be fetched by SAS is calculated (total # of hits / # of hits per page) and saved as macro variable &pg.

1.3 Next, a SAS macro loop is set up to load each result page based on the total number of page &pg as follows:

```

%let url=%nrstr(http://www.clinicaltrial.gov/ct2/results?cntry1=ES%3ACN&pg=);
%do i=1 %to &pg;
  filename fetch_&i url "&url.&i" debug lrecl=8192;
  data http_&i;
    infile fetch_&i length=len;
    input record $varying8192. len;
  run;

data _null_;
  wait=sleep(10);
run;
%end;

```

This macro will be executed for 73 times to download the contents of 73 pages of results. At each execution, an incremental page number is generated and appended to the end of URL for the result page. A same filename URL statement and the subsequent data step will then be invoked to act upon the newly generated URL to create SAS dataset for this web contents. One thing worth of note is that programmatic web fetch may cause excessive internet traffic on the targeted server. Many servers choose to terminate link or even block user's IP address to prevent this from happening. A good exercise from user's end is to reduce the intensity of data exchange by controlling the execution time. This can be accomplished by sleep() function within the data _null_ step, which automatically suspend SAS task for the number of seconds defined in the parentheses of sleep function after each macro execution. Finally the datasets for all result pages are set together for the future data processing.

```

proc sql noprint;
  select count(distinct memname) into: httpcnt
  from dictionary.columns where substr(memname,1,5)='HTTP_';
quit;
%put &httpcnt;

data all;
  set %do i=1 %to &httpcnt;
    http_&i(in=_&i)
    %end;;
  %do i=1 %to &httpcnt;
    if _&i then page=&i;
  %end;;
run;

```

1.4 Clean up the unformatted HTML text. Once all the relevant information are downloaded and dumped into data warehouse, the only thing left to do is to use SAS data step function again to locate and parse out the key parameters. For our example, clinical trial ID (NCT ID), the name of study and the study URL are the key elements that need to be extracted from the unformatted web contents. This requires user's familiarity with the contents and layout of target web pages. After reading the source text of the HTML page, we determine that these key elements are in the same HTML entity and can be located by unique linguistic pattern and special characters.

```

data allx;
  set all;
  if index(record,'<a title=') and index(record,'Show study NCT')
  then do;
    study=scan(record,2,'>');
    study=scan(study,1,'<');
    record=tranwrd(record,'href=', '@');
    url=scan(record,2,'@');
    url=scan(url,1,'>');
    url=compress(url,' ');
    id=scan(url,1,'?');
    id=scan(id,3,'/');
    output;
  end;
  keep study url id;
run;

```

As one can see in the above example, the linguistic pattern that characterizes the key elements are replaced with designated special characters by tranwrd() function and then the relevant data are parsed out by scan() function. The ready-to-use SAS dataset is shown in Figure 1d. This baseline dataset will be saved in a permanent place for the comparison with upcoming updated data.

Step 2: 24 hours later, use filename URL again to conduct the same internet search described in step 1 and save the resulting dataset as “updated” database.

Step 3: Compare the “updated” database from step 2 with the “baseline” dataset obtained in step 1 and the new records are considered to be the trials conducted in China that are newly registered during the period of time between step 1 and step 2.

```
proc sql noprint;
  select url into: urllist separated by '$' from updatex
  where study not in (select study from baseline);
  select count(url) into: urlcnt from updatex
  where study not in (select study from baseline);
quit;
```

In this step, the URLs for the newly registered trials within the past 24 hours are written into a macro variable &urllist.

Step 4: Use a SAS macro to sequentially retrieve the detailed trial information by filename URL based on the list of web URLs obtained from step 3.

```
data newurl; delete; run;
%if &urlcnt %then %do i=1 %to &urlcnt;
  %let url=%qscan(%nrquote(&urllist),&i,$);

filename fetch url "http://www.clinicaltrial.gov/&url" debug lrecl=8192;
  data new;
    infile fetch length=len;
    input record $varying8192. len;
  run;
data _null_;
  wait=sleep(5);
run;
%end;
```

In this step, the detailed information about the new trials is retrieved with filename URL statement by the approach described in step 1 and saved into a SAS dataset new.

Step 6: An automated email alert is triggered whenever a trial is identified to meet the user specification (location=China; sponsor=XXXX; condition=XXXXXX).

```
data newurl;
  set newurl new(in=a);
  length line $256.;
  retain flg;
  if not a then output;
  if a then do;
    temp=lag(record);
    if index(lowercase(record), 'wyeth') and
       index(temp, 'Sponsor:') then flg=1;
    if flg and index(lowercase(record), 'abdominal abscess')
       and index(temp, 'class="body3"')
    then do;
      line="&study"; output;
      line="http://www.clinicaltrial.gov/&url"; output;
      line=""; output;
    stop;
  end;
end;
keep line;

run;
```

Since we already use pre-defined URL to fetch the trials conducted in China, now we only need to search for the name of sponsor and disease condition that match our target criteria. Because these pieces of information are located in the different lines of HTML text, their co-appearance in a single study page is not particularly easy to be identified by SAS data step which reads textual contents line-by-line. In addition, the position of relevant sponsor and disease information need to be confirmed by the preceding textual pattern. For instance, the line before the name of trial sponsor must contain the string "Sponsor:" and the line with disease information is preceded by the text string 'class="body3"'. Here we use lag() function to store the value preceding the current observation into a new variable temp. If both the current observation contains the expected company name and preceding line captured by lag() function contains the string "Sponsor:" a retained variable "flag" is assigned to 1 for the current and upcoming observations to indicate that the current trial is sponsored by the company of interest. Subsequently, if the line that immediately follows the string 'class="body3"' carries the symptom name under scrutiny given that the trial is conducted by the sponsor of interest (flag=1), this trial is determined to be relevant so that its trial title and URL is written to the result dataset "newurl" from which the surveillance can be disseminated to the end-users or transferred to data warehouse.

Readers can use the following link as guidance to set up email server correctly in order to receive the email alert for the newly registered trial.

<http://support.sas.com/kb/19/767.html>

The following SAS code detects the existence of URL which triggers the email alert:

```
proc sql noprint;
    select count(line) into: newcnt from newurl;
quit;

%if &newcnt %then %do;
filename mymail email "your.emailaddress.com_or_edu" subject="Newly Registered
Clinical Trial";
data _null_;
    file mymail;
    put line;
run;
quit;
%end;
```


Step 7: The whole program is configured in batch mode so that the process can be launched at midnight of every day.

The configuration of SAS batch mode for scheduled task can be found in the following link:

<http://support.sas.com/techsup/technote/ts648/ts648.pdf>

DISCUSSION

This paper demonstrates the utilization of SAS as a simple web browser as opposed to its more widely appreciated roles as a powerful data manipulation and statistical package. In fact it is quite straightforward to use filename URL statement to fetch the web textual contents. The only caution here is to avoid overloading remote server by postponing SAS web fetching with sleep() function. The rich collection base SAS functions such as index(), tranwrd() and scan() also make easy parse out relevant information from the relatively complicated HTML textual structure. One caveat here is that HTML text may contain some SAS macro reserved special characters such as & and %, which poses quite a challenge when these characters are to be passed to SAS MACRO language. In our example we use MACRO quoting function %nrquote() to prevent these special characters from being interpreted in a "SAS MACRO" way. This is especially important when a web URL is to be written into a MACRO variable because it may even contain ";" which indicates the end of one SAS statement.

Our SAS tool described here represent a primitive form of more advanced feed function (RSS ) or email alert system that are offered by many websites nowadays. The real power of this SAS based web search, however, is that it can be integrated into the existing SAS data analysis pipeline or configured to channel the data from various sources into one single repository within a universal framework. Meanwhile, one must recognize that URL access of filename statement only works for those relatively simple web sites which do not requires registered user login. It is also important to be mindful of copy right issues when ones attempt to programmatically extract information from proprietary data source protected by user login. For more information on the web access by SAS, interested readers can further read ref [3]. In fact, some of our sample codes for URL access are simply copied from this reference.

REFERENCES

- 1 How to send SMTP email with SAS <http://support.sas.com/kb/19/767.html>
- 2 Examples of Batch Processing under Windows <http://support.sas.com/techsup/technote/ts648/ts648.pdf>
- 3 Garth Helf, Extreme Web Access: What to Do When FILENAME URL Is Not Enough. SUGI 30 Proceeding

CONTACT INFORMATION

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

Name: Xin Wei
Enterprise: Hoffmann-La Roche
Address: 340 Kingsland Street
City, State ZIP: Nutley, NJ, 07110
Work Phone: 973-235-2520
Fax: 973-235-2134
E-mail: xinwei@stat.psu.edu
Web:

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.