

Paper 4658-2020

From Device Text Data to a Quality Dataset

Laurie Smith, Cincinnati Children's Hospital Medical Center

ABSTRACT

Data quality in research is important. It may be necessary for data from a device to be used in a research project. Often it is read manually from an external file and entered onto a CRF. Then the data is manually read from the CRF and entered it into a database. This process introduces many opportunities for data quality to be compromised. The quality of device data used in a study can be greatly improved if the data can be read directly from a device's output file directly into a dataset. If the device outputs results into a file that can be saved electronically, SAS® can be used to read the data needed from the results and save the data directly into a dataset.

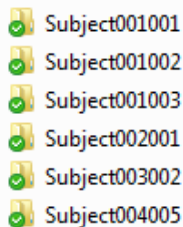
Quite often, device data is saved in separate files per subject and it is often difficult to import each separate file into SAS without great effort. If data is organized with the subject ID as the folder name and each subject's data in the corresponding folder, SAS® can also be used to read the data from a general location, importing all data within each location. In addition to improving data quality, data collection and monitoring time can also be reduced by taking advantage of these electronic files as opposed to recapturing this data on a CRF.

INTRODUCTION

The methods introduced will allow a SAS® Programmer, with basic SAS® programming skills, use SAS® to use SAS® to import individual subject data files saved with similar files name and folder name structures and import desired test results from a report in Excel® or fixed text format generated by a testing device.

SUBJECT DATA

Very often, data files or test results, generated by a device (MRI reads, Catheterization and other types of blood testing), are created and saved in separate directories with similar names per subject and similarly named data files per subject. All this data usually needs to be imported into one dataset.



A screenshot of a directory listing showing six subject folders. Each folder name is preceded by a green checkmark icon. The folder names are: Subject001001, Subject001002, Subject001003, Subject002001, Subject003002, and Subject004005.

- Subject001001
- Subject001002
- Subject001003
- Subject002001
- Subject003002
- Subject004005

Figure 1. Example of Subject Data Directory**IMPORT LIST OF SUBJECTS**

When data is organized such that there is one file per subject in a directory, and there is no set list of subjects, it is simple to obtain a listing of subjects to use for importing all data.

USE A FILENAME STATEMENT TO CREATE THE LIST OF SUBJECTS

A filename statement using pipe and a dir command can be used with an infile dataset to obtain a listing of the folder names from the desired directory. Using a folder naming convention that includes the subject's identifier allows a user to isolate each identifier and create a listing of all identifiers from the directory in one dataset.

```
filename dir pipe 'dir "<Source Data Location>" ';
```

```
data dirlistxx;
    infile dir lrecl=32727 truncover scanover;
    input dirtxt $200.;
run;
```

dirtxt	
Volume in drive C is OSDisk	
Volume Serial Number is 8CCC-DBD4	
Directory of C:\Users\BISE3R\OneDrive - cchmc\MWSUG2018\TestData	
08/10/2018 05:49 PM	<DIR> .
08/10/2018 05:49 PM	<DIR> ..
08/24/2018 10:40 PM	<DIR> Subject001001
08/24/2018 10:38 PM	<DIR> Subject001002
08/24/2018 10:37 PM	<DIR> Subject001003
08/24/2018 10:35 PM	<DIR> Subject002001
08/24/2018 10:32 PM	<DIR> Subject003002
08/24/2018 10:34 PM	<DIR> Subject004005
0 File(s)	0 bytes
8 Dir(s)	281,374,351,360 bytes free

Figure 2. dirlistxx

SUBSET SUBJECT IDS TO IMPORT DATA

Once the dataset with the folder names has been obtained, the subset of the data is created to exclude any observations that do not contain subject identifiers. A substring of each folder name isolating each identifier can be created in a separate variable. Only this new variable is retained. The count of subject identifiers is retained as a macro variable for the loop in the macro that imports the data.

```
data dirlist (drop=dirtxt);
    length subjid $500.;
    set dirlistxx;
if dirtxt^='' and scan(dirtxt,1,'')^='Volume' and
    scan(dirtxt,1,'')^='Directory' and scan(dirtxt,2,'')
    ^in('File(s)', 'Dir(s)') and scan(dirtxt,-1,'') ^in("..",".");
subjid=substr(scan(dirtxt,-1,''),8);
run;
```

```
data dirlist;
    set dirlistx nobs=subjcnt;
call symput('subjcnt',put(subjcnt,best12.));
run;
proc sort data=dirlist; by subjid; run;
```

subjid
001001
001002
001003
002001
003002
004005

Figure 3. dirlist

IMPORT SUBJECTS' DATA INTO ONE DATASET

USE A FILENAME STATEMENT TO CREATE THE LIST OF DATA FILES

Similar to above, a listing of results data filenames can be created from each subject's folder to use for importing the data into one dataset. A filename statement using pipe and a dir command can be used with an infile dataset so create a listing of all files in each subject's folder. A dataset of this list with only filenames is then retained.

```
%macro Panel(obsnum=);
  %do i=1 %to &obsnum;
    data _null_;
      set dirlist end=eof;
    if _n_=&i then do;
      call symput('subjid',strip(trim(subjid)));
    end;
  run;

  %let subjData=&Source.\Subject&subjid;
  libname subjData "&subjData.";
  %let subjData=%sysfunc(quote(%qsysfunc(dequote
    (&subjData))));
  filename sdir&i pipe %sysfunc(quote(dir &subjData));

  data subjdirXX&i;
    infile sdir&i lrecl=32727 truncover scanover;
    input subjdir $200.;
  run;

  data subjdirX&i (keep=panel);
    set subjdirXX&i;
    if subjdir^='' and scan(subjdir,1,'')^='Volume' and
      scan(subjdir,1,'')^='Directory' and scan(subjdir,2,'')
      ^in('File(s)','Dir(s)') and scan(subjdir,-1,'')
      ^in(' ','..');
    panel=cat(scan(subjdir,-3,'. '),'',scan(subjdir,-2,'. '));
  run;

  data subjdir&i;
    set subjdirX&i nobs=panelcnt;
    call symput('panelcnt',put(panelcnt,best12.));
  run;
```

panel
Panel 1
Panel 2
Panel 3

Figure 4. subjdir1 (data files for first subject)

EXTRACTING DATA FROM EXCEL® SPREADSHEET REPORTS GENERATED BY A DEVICE

IMPORT DATA INTO ONE DATASET

A do loop is used to loop through each subject id to create a dataset of each subject's files to be imported by storing the id in a macro variable (as seen above). Once the subject's id is defined in the do loop, a second do loop is executed to create a dataset containing filenames of files to be imported in the final dataset.

```
%do j=1 %to &panelcnt;
    data _null_;
        set subjdir&i;
    if _n_=&j then do;
        call symput('panel',trim(strip(panel)));
        call
            symput('comppanel',trim(strip(compress(panel))));
    end;
run;

proc import out=subject&subjid.&comppanel.x
    datafile="&Source.\Subject&subjid.\&panel..xlsx"
        dbms=xlsx replace;
    sheet="&panel";
    getnames=yes;
run;

data subject&subjid.&comppanel;
    set subject&subjid.&comppanel.x;
    subjid="&subjid";
    panel="&panel";
run;

data panels;
    set panels
        subject&subjid.&comppanel;
    if subjid^='';
run;
%end;
%end;
%mend;
```

The structure of the final dataset is determined per the structure of the data in the imported files and once do loops are executed for all ids, a final dataset that contains all results from the imported files is created. In this example all files are of XLSX format and panel number and subject identifier are included in the final dataset.

	A	B	C	D	E	F
1	Test No	Result 1	Result 2	Result 3	Result 4	Result 5
2	Test 1	0.048	0.612	1.273	8.983	0.279
3	Test 2	0.237	0.721	1.463	9.683	0.168
4	Test 3	0.388	0.501	1.683	7.913	0.356

Figure 5. Source data structure

Execute the macro:

```
%Panel (obsnum=&subjcnt) ;
```

subjid	panel	Test_No	Result_1	Result_2	Result_3	Result_4	Result_5
001001	Panel 1	Test 1	0.025	0.589	1.25	8.96	0.256
001001	Panel 1	Test 2	0.214	0.698	1.44	9.66	0.145
001001	Panel 1	Test 3	0.365	0.478	1.66	7.89	0.333
001001	Panel 2	Test 1	0.118	0.682	1.343	9.053	0.349
001001	Panel 2	Test 2	0.307	0.791	1.533	9.753	0.238
001001	Panel 2	Test 3	0.458	0.571	1.753	7.983	0.426
001001	Panel 3	Test 1	0.228	0.792	1.453	9.163	0.459
001001	Panel 3	Test 2	0.417	0.901	1.643	9.863	0.348
001001	Panel 3	Test 3	0.568	0.681	1.863	8.093	0.536
001002	Panel 1	Test 1	0.201	0.765	1.426	9.136	0.432
001002	Panel 1	Test 2	0.39	0.874	1.616	9.836	0.321
001002	Panel 1	Test 3	0.541	0.654	1.836	8.066	0.509
001002	Panel 3	Test 1	0.193	0.757	1.418	9.128	0.424
001002	Panel 3	Test 2	0.382	0.866	1.608	9.828	0.313
001002	Panel 3	Test 3	0.533	0.646	1.828	8.058	0.501
001003	Panel 1	Test 1	0.135	0.699	1.36	9.07	0.366
001003	Panel 1	Test 2	0.324	0.808	1.55	9.77	0.255
001003	Panel 1	Test 3	0.475	0.588	1.77	8	0.443
001003	Panel 2	Test 1	0.099	0.663	1.324	9.034	0.33
001003	Panel 2	Test 2	0.288	0.772	1.514	9.734	0.219
001003	Panel 2	Test 3	0.439	0.552	1.734	7.964	0.407
002001	Panel 2	Test 1	0.128	0.692	1.353	9.063	0.359
002001	Panel 2	Test 2	0.317	0.801	1.543	9.763	0.248
002001	Panel 2	Test 3	0.468	0.581	1.763	7.993	0.436
002001	Panel 3	Test 1	0.168	0.732	1.393	9.103	0.399
002001	Panel 3	Test 2	0.357	0.841	1.583	9.803	0.288
002001	Panel 3	Test 3	0.508	0.621	1.803	8.033	0.476
003002	Panel 1	Test 1	0.071	0.635	1.296	9.006	0.302
003002	Panel 1	Test 2	0.26	0.744	1.486	9.706	0.191
003002	Panel 1	Test 3	0.411	0.524	1.706	7.936	0.379
004005	Panel 3	Test 1	0.048	0.612	1.273	8.983	0.279
004005	Panel 3	Test 2	0.237	0.721	1.463	9.683	0.168
004005	Panel 3	Test 3	0.388	0.501	1.683	7.913	0.356

Figure 6. Result of macro execution (All panel data from all subjects)

EXTRACTING DATA FROM TEXT REPORTS GENERATED BY A DEVICE

DEVICE DATA AS A TEXT FILE

Some devices generate text files containing results from testing. Each file is usually presented in a fixed format with one file per subject per test.

Test 2 RESULTS REPORT

Generated: 8/11/2017 1:25:03 PM

PATIENT INFORMATION

Patient Initials: LB
Patient ID: 0101
Patient gender: F
Birth date: 2/11/1973
Patient weight: 91 kg
Patient height: 178 cm
BSA: 2.12 m2
Heart rate: 88 bpm

SUMMARY

Test Result 1: 11.12 cm/s
Test Result 2: 1.70 mmHg
Test Result 3: 801.00 ms
Test Result 4: 2.42 l/min
Test Result 5: 0.00 %
Test Result 6: 33.62 ml/beat
Test Result 7: 0.00 l/beat

Figure 7. Example of Device data text file

The results may contain several results for a subject grouped by a particular characteristic.

Test 1 RESULTS REPORT

Generated: 8/11/2017 1:22:35 PM

PATIENT INFORMATION

Patient Initials: LB
Patient ID: 0101
Patient gender: F
Birth date: 2/11/1973
Patient weight: 91 kg
Patient height: 178 cm
Heart rate: 88 bpm

LEFT REGION RESULTS

Body Surface Area: 2.12 m2
Test Result 1: 10.44 ml
Test Result 2: 7.54 ml/m2
Test Result 3: 8.33 ml
Test Result 4: 6.21 ml/m2
Test Result 5: 157.93 ml/min
Test Result 6: 20.05 %
Test Result 7: 19.27 g
Test Result 8: 14.02 g/m2
Test Result 9: 12.61 g/m

RIGHT REGION RESULTS

Body Surface Area: 2.12 m2
Test Result 1: 142.00 ml
Test Result 2: 103.62 ml/m2
Test Result 3: 72.55 ml
Test Result 4: 53.28 ml/m2
Test Result 5: 5.12 l/min
Test Result 6: 48.47 %
Test Result 7: 65.61 g
Test Result 8: 47.51 g/m2
Test Result 9: 43.10 g/m

Figure 8. Example of Device data text file

USING SAS® TO EXTRACT DESIRED DATA

The process of manually entering this data onto a CRF, then into a database can be eliminated if SAS® is used to extract desired data directly from the text report.

It is best to use this program in tandem with the macro above to obtain the data filenames, since data for multiple subjects will most likely be imported.

Extraction from a simple report

Extracting data from a report as seen in Figure 8 involves reading the txt file in a data step using an infile statement, starting at line 3 (where the data starts). For this extraction, only the testing date (Generated), Patient Initials, Subject ID (Patient ID), Test Heart Rate (Heart rate), Test Result 6, and Test Result 7 are needed for the final dataset. The subject ID is assigned to a macro variable to serve as an identifier for later use in the macro. The data step finds the line containing the desired data value, reads in the value and units and stores it in a character variable.

```
%macro Tst2(filename);
  data Tst2x;
    infile "(Data Location)\&filename..txt" firstobs=3 trunccover
          scanover;
    input
      @'Generated:' itstdt $100.
      @'Patient Initials:' iinit $100.
      @'Patient ID:' isubjid $100.
      @'Heart rate:' itst2hr $100.
      @'Test Result 6:' itstrslt6 $100.
      @'Test Result 7:' itstrslt7 $100.;
    call symput('subjid',strip(trim(isubjid)));
  run;
```

The next data step in the macro creates a subject specific dataset that isolates the numeric values from each character test result variable from the dataset above (dropping the character variables).

```
data tst2&subjid. (drop=iinit itstdt isubjid itstrslt6 itstrslt7 itst2hr);
  retain subjid;
  length subjid $25. init $3.;
  format tstdt date9.;
  set tst2x;
  tstdt=input(scan(itstdt,1,' '),mmddy10.);
  subjid=strip(trim(isubjid));
  iinit=strip(trim(iinit));
  tst2hr=input(scan(itst2hr,1,' '),best12.);
  tst2rslt6=input(scan(itstrslt6,1,' '),best12.);
  tst2rslt7=input(scan(itstrslt7,1,' '),best12.);
  run;
  proc sort data=tst2&subjid; by subjid tstdt; run;
```

```
%mend;
```

Call the macro with the filename as the parameter:

```
%Tst2(Subj 0101 LB Test 2);
```

Resulting in the following output:

subjid	init	tstdt	tst2hr	tst2rslt6	tst2rslt7
0101	LB	11AUG2017	88	33.62	0

Figure 9. Test 2 Result

Extraction from a grouped report

Extracting data from a report as seen in Figure 9 is similar to the above extraction except data must be extracted per section. The data values needed for the final dataset are the testing date (Generated), Patient Initials, Subject ID (Patient ID), Test Heart Rate (Heart rate), Left Region Test Result 1, Left Region Test Result 3, Left Region Test Result 6, Left Region Test Result 8, Right Region Test Result 1, Right Region Test Result 3, Right Region Test Result 6, and Right Region Test Result 8.

Since the test descriptions for the Left and Right Regions are the same, they will have to be extracted in separate data steps. The identifiers (testing date, Patient Initials, and Subject ID) will be extracted for Left and Right Regions such that the Left and Right Region data can be merged to create one dataset. Test Heart Rate will be extracted with the Left Region data only.

Left Region Extraction

```
%macro Tst1lft(filename);
  data Tst1lftx;
    infile "(Data Location)\&filename..txt" firstobs=3 truncover
      scanover;
    input
      @'Generated:' itstdt $100.
      @'Patient Initials:' iinit $100.
      @'Patient ID:' isubjid $100.
      @'Heart rate:' itstlhr $100.
      @'LEFT' lft $100.;
    if trim(lft)='REGION RESULTS' then do;
      input
        @'Test Result 1:' itstrslt1 $100.
        @'Test Result 3:' itstrslt3 $100.
        @'Test Result 6:' itstrslt6 $100.
        @'Test Result 8:' itstrslt8 $100.;
    end;
    call symput('subjid',strip(trim(isubjid)));
run;
```

Call this macro with the filename as the parameter:

```
%Tst1rgt(Subj 0101 LB Test 1);
```

Resulting in the following output:

subjid	init	tstdt	tst1hr	lftst1rslt1	lftst1rslt3	lftst1rslt6	lftst1rslt8
0101	LB	11AUG2017	88	10.44	8.33	20.05	14.02

Figure 10. Test 1 Left Region Result

Right Region Extraction macro

```
%macro Tst1rgt(filename,subjid);
  data Tst1rgtx;
    infile "(Data Location)\&filename..txt" firstobs=3 truncover
```



```

        scanover;
input
    '@Generated:' itstdt $100.
    '@Patient Initials:' iinit $100.
    '@Patient ID:' isubjid $100.
    '@RIGHT' rgt $100.;
    if trim(rgt)='REGION RESULTS' then do;
        input
            '@Test Result 1:' itstrslt1 $100.
            '@Test Result 3:' itstrslt3 $100.
            '@Test Result 6:' itstrslt6 $100.
            '@Test Result 8:' itstrslt8 $100.;
        end;
    call symput('subjid',strip(trim(isubjid)));
run;

data Tst1rgt&subjid. (drop=itstdt iinit isubjid rgt itstrslt1 itstrslt3
                    itstrslt6 itstrslt8);
    retain subjid;
    length subjid $25. init $3.;
    format tstdt date9.;
    set Tst1rgtx;
    tstdt=input(scan(itstdt,1,' '),mmdyy10.);
    subjid=strip(trim(isubjid));
    iinit=strip(trim(iinit));
    rgttst1rslt1=input(scan(itstrslt1,1,' '),best12.);
    rgttst1rslt3=input(scan(itstrslt3,1,' '),best12.);
    rgttst1rslt6=input(scan(itstrslt6,1,' '),best12.);
    rgttst1rslt8=input(scan(itstrslt8,1,' '),best12.);
run;
proc sort data=Tst1rgt&subjid; by subjid init tstdt; run;
%mend;

```

Call this macro with the filename as the parameter:

```
%Tst1rgt(Subj 0101 LB Test 1);
```

Resulting in the following output:

subjid	init	tstdt	rgttst1rslt1	rgttst1rslt3	rgttst1rslt6	rgttst1rslt8
0101	LB	11AUG2017	142	72.55	48.47	47.51

Figure 11. Test 1 Right Region Result

Create final dataset

All extracted data for the subject can now be merged into one dataset. Again, defined as a macro where the parameter is the Subject ID, in order to more easily merge all data per subject.

```

%macro mergeDData(subjid);
    data DeviceTextData&subjid;
        merge tst1lft&subjid.
            tst1rgt&subjid.
            tst2&subjid.;

```

```

    by subjid init tstdt;
label subjid='Subject ID'
    init='Subject Initials'
    tstdt='Test Date'
    tst1hr='Subject Test 1 Heart Rate'
    lfttst1rslt1='Left Region Test 1 Result 1'
    lfttst1rslt3='Left Region Test 1 Result 3'
    lfttst1rslt6='Left Region Test 1 Result 6'
    lfttst1rslt8='Left Region Test 1 Result 8'
    rgttst1rslt1='Right Region Test 1 Result 1'
    rgttst1rslt3='Right Region Test 1 Result 3'
    rgttst1rslt6='Right Region Test 1 Result 6'
    rgttst1rslt8='Right Region Test 1 Result 8'
    tst2hr='Subject Test 2 Heart Rate'
    tst2rslt6='Test 2 Result 6'
    tst2rslt7='Test 2 Result 7';

run;
%mend mergeDData;

```

Call this macro with the subjid as the parameter:

```
%mergeDData(0101);
```

Resulting in the following output:

Subject ID	Subject Initials	Test Date	Subject Test 1 Heart Rate	Left Region Test 1 Result 1	Left Region Test 1 Result 3	Left Region Test 1 Result 6	Left Region Test 1 Result 8	Right Region Test 1 Result 1	Right Region Test 1 Result 3	Right Region Test 1 Result 6	Right Region Test 1 Result 8	Subject Test 2 Heart Rate	Test 2 Result 6	Test 2 Result 7
0101	LB	11AUG2017	88	10.44	8.33	20.05	14.02	142	72.55	48.47	47.51	88	33.62	0

Figure 12. Final dataset for subject

CONCLUSION

In some cases, it is necessary to collect subject data one file per subject per result. Storing each subject's data using the same naming convention for the files and folders in one folder per subject allows for easier import into one dataset. One of the macros presented above can be used to import this data.

It is often practice to transcribe the results needed for research from device results in txt format from the text reports onto a CRF, where the data on the CRF are later entered into a database. Using SAS® to read the desired data values directly from the text report into a dataset can eliminate the need for this process.

This process can easily be adjusted to retain units for value conversions to one standard unit in case results provided are presented in different units.

CONTACT INFORMATION

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

Laurie A Smith

Cincinnati Children's Hospital Medical Center
(513) 803-9001
laurie.bishop@cchmc.org
www.cincinnatichildrens.org

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.