

The Converter Tool: an ETL solution built with SAS/AF[®]

Béco Isia, Merck Sharp & Dohme, Brussels, Belgium

ABSTRACT

In order to meet the needs of a study that Merck Sharp & Dohme (MSD) conducted, an in-house ETL solution called the Converter Tool has been developed using SAS/AF[®]. This tool helps Lab Services collect and translate external data provided in SPSS to a tab-delimited ASCII file compatible with the generic format of the Lab repository. In this paper the features of this tool and its data processing principles are reviewed and in particular how it processes cumulative data files.

INTRODUCTION

In 2004, MSD started a study requiring the integration of external data to the Lab repository. The data collected include the patient visit, demographic information and ECG test results. The data were delivered in the form of cumulative SPSS data files. MSD faced a problem of data integration since the External Data Management System (EDMS), a Visual Basic tool built for this purpose, could not handle SPSS files and had not a mechanism to process cumulative files. In fact EDMS considers any revised record from subsequent cumulative files like a new record and therefore does not override the old record. To bridge the gap, a SAS/AF tool called the Converter Tool has been developed. This tool brought, through its series of SAS steps and SAS SCL programs, a critical solution in:

- converting SPSS files to tab-delimited ASCII file
- removing duplicate data
- performing revision control
- providing the necessary means in term of audit trail to meet 21 CFR Part 11 requirements by tracing the modifications to every record loaded to the Lab repository and using the electronic signature.

The flowchart (figure 1) describes the data flow and provides a logical diagram of how the system operates.

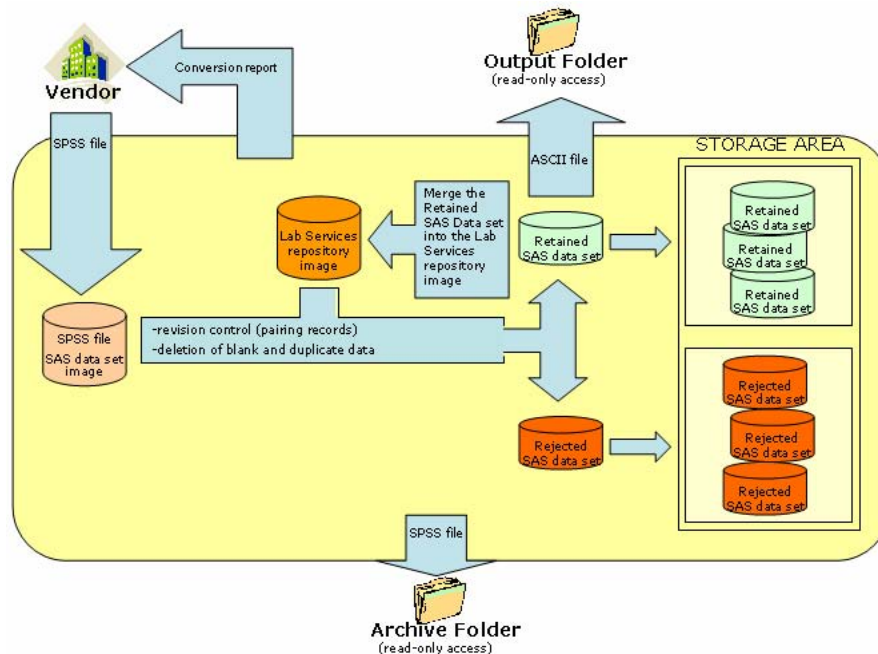


Figure 1
CONVERTER TOOL FLOWCHART

INSTALLING THE CONVERTER TOOL

REQUIREMENTS

The Converter Tool requires the following products:

- Windows NT or higher
- SAS for Windows v8 or higher
- SPSS for Windows 11.0 or higher
- Crystal Report Viewer
- Windows notepad
- Acrobat PDF Writer
- Microsoft Excel

SETUP

The first step is to create a directory in which the tool will reside along with the Crystal Reports files which serve like report generators for reporting the conversion output and log. This directory will store a SAS dataset that reflects the current status of the Lab repository, the retained dataset for each converted SPSS file, the rejected data and the information regarding the converted files. Additionally an Excel file named "CRprint.xls" should be declared as an ODBC data source for all Crystal Report files.

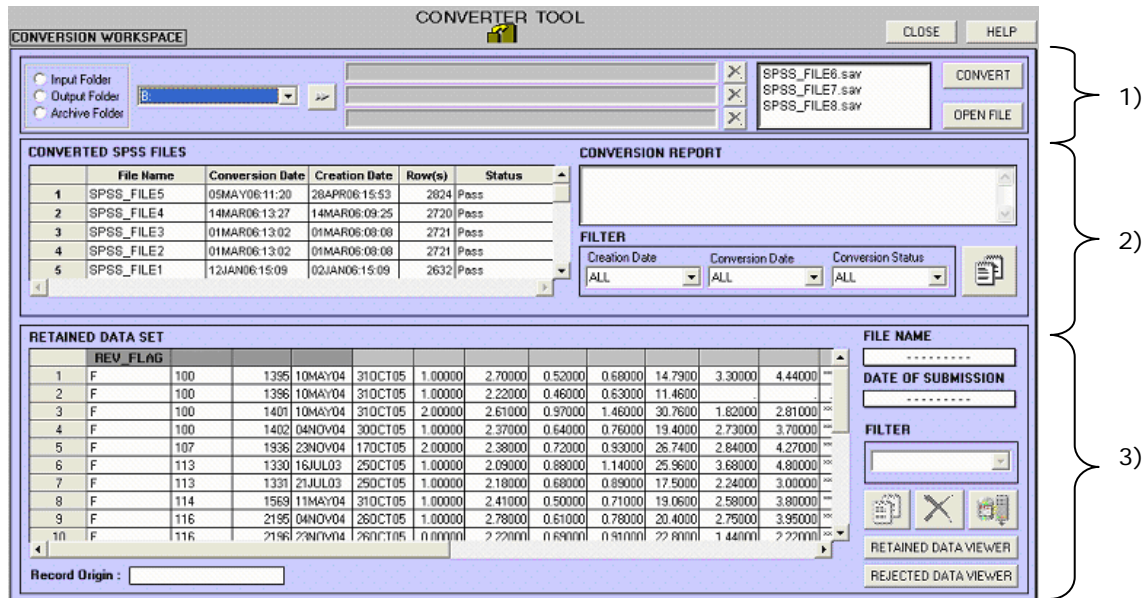
SECURITY

The access to the Converter Tool is available only to authorized users and during a session, the tool is locked by the current user.

FEATURES

CONVERSION WORKSPACE

When a session is opened, the Conversion Workspace screen is displayed. This main interactive SAS/AF frame performs a series of validation checks that can cause either the failure of the conversion or the revision of the data. It is divided in three parts (figure 2):



**Figure 2
CONVERSION WORKSPACE FRAME**

1) The first part is used to open an SPSS file in 'Read-Only' access and to convert the selected SPSS files. It can also be used to set the path where the incoming SPSS files are stored (Input Folder), the path where these files are stored after the conversion (Archive Folder) and the path where the ASCII data files are collected (Output Folder). The list box (on the right) displays the SPSS files found once the "Input Folder" is set. A simplified SCL code used to list these files is shown below:

```

rc=filename('mydir', Input_Text.text);
dirid=dopen('mydir');
name='';
count=0;
memcount=dnum(dirid);
files_list=makelist();
do j=1 to memcount;
  name=dread(dirid,j);
  if upcase(substr(name,(index(name,'.')+1))) = 'SAV' then do;
    rc=insertc(files_list,lstname,-1);
    count = count + 1;
  end;
end;
rc=dclose(dirid);
ListBox.items=file_list;
Rc=dellist(file_list)

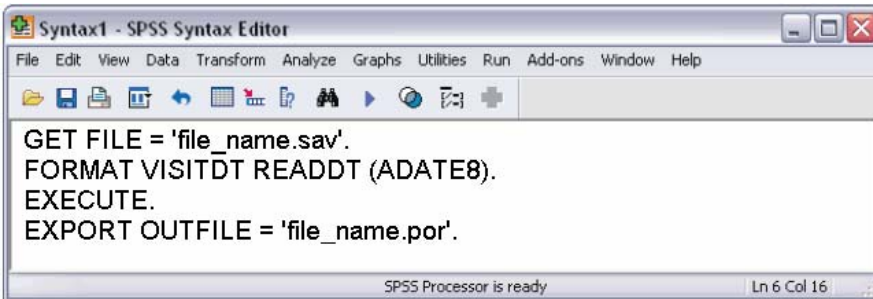
```

Where:

Input_Text is the text field for the "Input Folder";

ListBox is the list box where the SPSS files are displayed.

The conversion is performed by clicking on the "CONVERT" button. As SAS can only read data originally saved in the form of an SPSS Export file (with the extension ".por"), the process starts with the translation of the native SPSS file (with the extension ".sav") to the portable format. The Converter Tool writes programmatically an SPSS script file (figure 3) then calls the SAS X command to execute this script file in order to generate the SPSS portable format.



```

Syntax1 - SPSS Syntax Editor
File Edit View Data Transform Analyze Graphs Utilities Run Add-ons Window Help
GET FILE = 'file_name.sav'.
FORMAT VISITDT READD (ADATE8).
EXECUTE.
EXPORT OUTFILE = 'file_name.por'.
SPSS Processor is ready Ln 6 Col 16

```

Figure 3
SPSS SCRIPT FILE

Below the SCL statement for writing a SPSS script file (with the extension ".sps").

```

/* The SPSS commands are in bold */
selected_items= ListBox.selecteditems;
do j=1 to ListBox.selectedcount;
  select_val = getitemc(selected_items,j);
  script_val = 'GET FILE =
  ''''||Input_Text.text||select_val||''''.';
  sysrc=DELETE('file_path\export.sps','file');
  submit;
  data _null_;
  file 'file_path\export.sps';
  put '&script_val';
endsubmit;
  script_val = 'FORMAT VISITDT READD (ADATE8).';
  submit;
  put '&script_val';
endsubmit;
  script_val = 'EXECUTE.';
  submit;
  put '&script_val';
endsubmit;
  pos=index(upcase(select_val),'.SAV')-1;
  script_val= 'EXPORT OUTFILE=

```

PhUSE 2006

```

'''file_path' || substr(select_val,1,pos) || '.por'''.';
submit continue;
  put '&script_val';
run;
endsubmit;
end;

```

Once the portable SPSS file is created, the following code is executed to generate a SAS dataset.

```

filename myfile 'file_path\file_name.por';
proc convert spss=myfile out= myoutput;
run;

```

2) The second part displays the attributes of the converted files along with the electronic signature of the user and the conversion log in the "CONVERSION REPORT" text box. To visualize the SPSS file in 'Read-Only' access, the user has to double-click on the corresponding record. The background color of the grid changes according to the conversion status (figure 4). White color indicates the conversion has been successfully completed, yellow that the conversion has been successfully completed but there is/are unknown column(s) or some records with missing identifiers (Primary Key) and red indicates the conversion failed because there is either a violation of the file naming convention or there is at least one missing ECG test column.

CONVERTED SPSS FILES						
	File Name	Conversion Date	Creation Date	Row(s)	Status	UserID
1	SPSS_FILE5	05MAY06:11:20	28APR06:15:53	2824	Pass	user 1
2	SPSS_FILE4	14MAR06:13:27	14MAR06:09:25	2720	Fail	user 2
3	SPSS_FILE3	01MAR06:13:02	01MAR06:08:08	2721	Pass	user 3
4	SPSS_FILE2	01MAR06:13:02	01MAR06:08:08	2721	Pass	user 4
5	SPSS_FILE1	12JAN06:15:09	02JAN06:15:09	2632	Pass	user 5

Figure 4
CONVERTED SPSS FILES DATAGRID

Let us look at an example of the code used to generate the dataset attached to this grid.

```

%let macdate;
%let macrow;

/*
SPSS_FILE in the statement below is the fileref. The PIPE option
is used to read the properties of the SPSS file.
*/

filename spss_file pipe 'dir file_path';

/*
The following code shows a DATA _NULL_ step that collects the
creation date of the SPSS file in datetime13 format then stores it
in the macro variable named MACDATE.
*/

data _null_;
infile spss_file trunccover ;
input @3 marker $1. @1 text $200.;
if marker='/' then do;
  info=scan(text,1,' ');
  month = substr(info,1,2);
  day = trim(substr(info,4,2));
  year = trim(substr(info,9,2));
  time=trim(scan(text,2,' '));
  size=right(scan(text,3,' '));
  if size='PM' then
  time=(input(substr(time,1,2),2.)+12)||':'||substr(time,4,2);
  date=trim(day)||trim(month)||trim(year)||':'||left(time);

```

```

call symput ('macdate',date);
output;
end;
run;

/*
The PROC SQL counts the number of rows and stores the result in
the macro variable named MACROW. MYOUTPUT is the dataset generated
from the SPSS file.
*/

proc sql noprint;
select count(*) into :macrow
from myoutput;
quit;

/*
The data step bellow gathers file information in a dataset.
MACFILENAME is a macro variable containing the file name and
MACSTATUS is a macro variable containing the conversion status
(PASS/FAIL).
*/

data attributes;
length status $1. file_name $20. userid $50.;
format conv_date create_date datetime13.;
file_name = "&macfilename";
conv_date= datetime();
create_date = input("&tampon",datetime13.);
row = &macrow;
status = "&macstatus";
userid = "&sysuserid";
run;

/*
A PROC APPEND is used to add the file information to the dataset
attached to the grid of the figure 3.
*/

proc append
base=files_attributes data= attributes;
run;

```

3) Figure 5A illustrates the retained dataset grid for each converted file. All new records matching the records that have been previously converted along with any blank records are not converted. The revised records are paired and have a green background color. By selecting a record, the "Record Origin" field displays the name of the converted file where the record originates from (figure 5B). An additional column called 'REV_FLAG' is added to the output in order to track the revision status: 'F' for new records, 'B' for expired and revised records, 'G' for non-expired and revised records, and '-' for duplicate new or revised records within the same file. Figure 6 depicts different outputs based on the different scenarios.

RETAINED DATA SET												FILE NAME		
	REV_FLAG												SPSS_FILE4	DATE OF SUBMISSION
1	B	100	1395	17APR03	26FEB04	0.00000	2.62000	0.50000	0.76000	16.2700	2.66000	4.09000		14MAR03
2	G	100	1395	17APR03	27FEB06	0.00000	2.76000	0.63000	0.70000	17.6400	2.95000	4.04000		
3	F	100	1401	10MAY04	31OCT05	2.00000	2.61000	0.97000	1.46000	30.7600	1.82000	2.81000		
4	F	100	1402	04NOV04	30OCT05	1.00000	2.37000	0.64000	0.76000	19.4000	2.73000	3.70000		
5	F	107	1936	23NOV04	17OCT05	2.00000	2.38000	0.72000	0.93000	26.7400	2.84000	4.27000		
6	F	113	1330	16JUL03	25OCT05	1.00000	2.09000	0.88000	1.14000	25.9600	3.68000	4.80000		
7	F	113	1331	21JUL03	25OCT05	1.00000	2.18000	0.68000	0.69000	17.5000	2.24000	3.00000		
8	F	114	1959	11MAY04	31OCT05	1.00000	2.41000	0.50000	0.71000	19.0600	2.58000	3.80000		
9	F	116	2195	04NOV04	26OCT05	1.00000	2.70000	0.61000	0.70000	20.4100	2.75000	3.95000		
10	F	116	2196	28NOV04	26OCT05	0.00000	2.70000	0.69000	0.91000	22.8600	1.44000	2.70000		

Record Origin :

Figure 5A
RETAINED DATASET GRID

Figure 5B
RETAINED DATASET GRID

INPUT	ACTION TAKEN	OUTPUT
Scenario1 : new record		
PK-XXX	The record is flagged with 'F'.	F-PK-XXX
Scenario2 : new records having the same identifier		
PK-XXX	The records are flagged with 'F'.	F-PK-XXX
PK-XXY		F-PK-XXY
Scenario3 : revised record		
PK-XXY	The old record is flagged with 'B' and the revised record is flagged with 'G'.	B-PK-XXX G-PK-XXY
Scenario4 : revised records having the same identifier		
PK-XXY	The old record is flagged with 'B' and revised records are flagged with 'F'.	F-PK-XXY
PK-XXZ		F-PK-XXZ B-PK-XXX

PK = identifier
 XXX } =Test fields
 XXY }
 XXZ }

Figure 6
CONVERTED SPSS FILES DATAGRID

THE RETAINED DATA VIEWER

The retained data viewer shows the records that have been sent to the Lab repository (figure 7). Here the revision for paired records (B/G) is performed on first-in-first-out basis. In fact the revised records newly loaded to the Lab repository are flagged with a 'G', the records previously flagged with 'G' will be flagged with a 'B' and the records flagged with a 'B' will be aged out.

Figure 7
RETAINED DATA VIEWER

THE REJECTED DATA VIEWER

Figure 8 displays the rejected records. The records could be rejected for one of the following reasons:

- within a file there are two (or more) new records having the same primary key (A)
- a record has a missing primary key (B)

	REV_FLAG	Allocation	Visit Date	Data File											
1	-	144	1807 22OCT03	24MAR05	0.00000	2.27000	0.71000	0.62000	23.6000	2.30000	3.18000	73.5500	4.03000	****	
2	-	144	1807 22OCT03	24MAR05	0.00000	2.27000	0.71000	0.62000	23.6000	2.30000	3.18000	73.5500	1 03000	****	(A)
3	-	25	3958												
4	-	29	3830												
5	-	29	3832												
6	-	44	1031 17JUN03	31MAR05	1.00000	2.32000	1.12000	1.32000	26.9400	1.79000	2.29000	48.1200	5.44000	****	
7	-	44	1031 17JUN03	30MAR05	1.00000	2.32000	1.12000	1.32000	26.9400	1.79000	2.29000	48.1200	5.44000	****	
8	-	85	2856 27NOV03	31MAR05	1.00000	2.31000	0.71000	0.90000	28.7000	2.72000	3.72000	86.6200	5.80000	****	
9	-	85	2856 27NOV03	31MAR05	1.00000	2.31000	0.71000	0.90000	28.7000	2.72000	3.72000	86.6200	5.80000	****	

Figure 8
REJECTED DATA VIEWER

CONCLUSION

The Converter Tool is a tailored SAS/AF ETL solution that can be generalized for reuse for other studies by adding parameters to setup specifications of the input file and by combining it with a data translator that can translate data to and from a large number of formats.

ACKNOWLEDGMENTS

The author would like to sincerely thank all who were available to give guidance and took time to review this manuscript. They brought valuable input based on their experiences. Please note that the codes supplied in this paper are designed only to illustrate the concepts of the tool and may need to be modified to work in other applications. The author of this paper does not support modified codes.

CONTACT INFORMATION

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

Béco Isia
 Merck Sharp & Dohme
 Clos de Lynx 5, 1200 Bruxelles
 Work Phone: +32 2 776 64 18
 Email: beco_isia_aroundala@merck.com
becoisia@yahoo.fr