

Paper PO-014

A SYSTEM TO STANDARDIZE CREATING FILES USING VISUAL BASIC 6.0 AND SAS® AUTOMATION

Mila Chiflikyan, Nick Kinsey, Jean Richardson, Donna Medeiros
(RTI International)

INTRODUCTION

RTI International conducts large surveys and studies, and obtains locating information about study participants from disparate sources; this causes the accuracy, completeness, and currency of the data to vary widely. To obtain better information, RTI submits files containing locating information on study participants to batch tracing vendors, who return files containing more accurate, complete, and current information. However, each batch tracing vendor has different requirements for content, format, and record layout for the files they receive. In addition, each study has unique data, file formats, and other parameters.

To save time and resources, RTI developed an efficient, user-friendly system that uses a Visual Basic 6.0 application to request information about the type of tracing needed and the study data available and passes the information to a SAS® application that formats, standardizes and generates the file to be submitted to the batch tracing vendor. The SAS application uses macros, import and export procedures, and text processing.

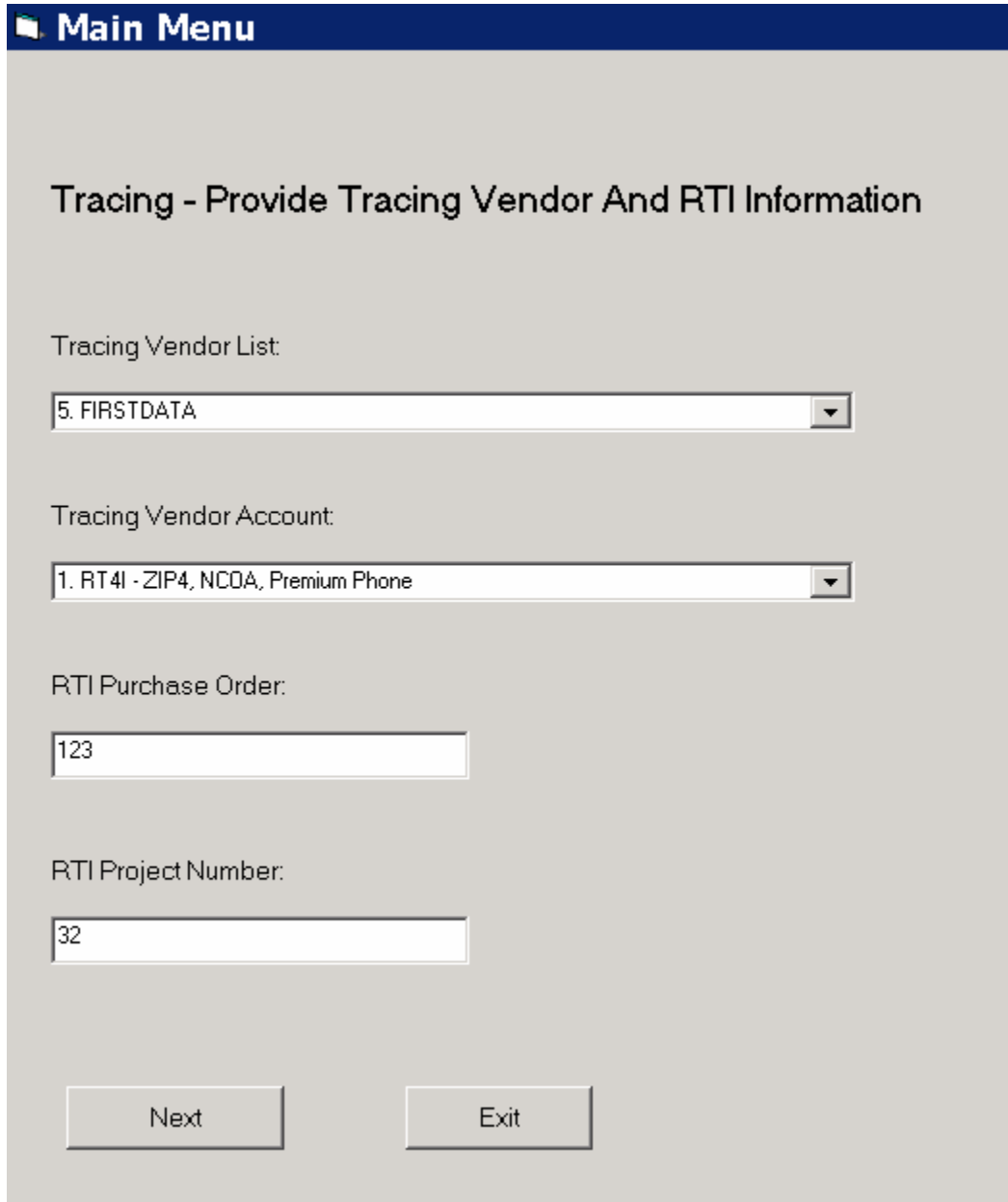
Users select the vendor they want to use, the file containing study data, and the data items in the file. The system then generates a file meeting the vendor's requirements along with a SAS log and list files. The system can accept as input SAS, Excel, comma delimited, and text files to generate files in the formats required by different batch tracing vendors.

The system is user friendly and does not require the user to know SAS or Visual Basic 6.0. With this system in place, project staff can quickly and easily generate files to be submitted to batch tracing vendors.

SYSTEM DESIGN OVERVIEW

To begin the process, project staff specify the type of tracing to be performed and provide billing information.

RTI has contracts with several batch tracing vendors. Each vendor offers several types of tracing and requires specific data items be submitted based on the type of tracing to be performed. As shown in Figure 1, projects select the vendor based on the type of tracing needed and the data available and the type of tracing to be performed from the list of vendor options. Projects also enter the purchase order the vendor will use to bill RTI for tracing and the account RTI will use to bill the project.



The screenshot shows a software window titled "Main Menu" with a dark blue header. Below the header, the main content area has a light gray background. The title "Tracing - Provide Tracing Vendor And RTI Information" is centered at the top of this area. There are four input fields arranged vertically: 1. "Tracing Vendor List:" with a dropdown menu showing "5. FIRSTDATA". 2. "Tracing Vendor Account:" with a dropdown menu showing "1. RT4I - ZIP4, NCOA, Premium Phone". 3. "RTI Purchase Order:" with a text box containing "123". 4. "RTI Project Number:" with a text box containing "32". At the bottom of the form, there are two buttons: "Next" and "Exit".

Main Menu

Tracing - Provide Tracing Vendor And RTI Information

Tracing Vendor List:

5. FIRSTDATA

Tracing Vendor Account:

1. RT4I - ZIP4, NCOA, Premium Phone

RTI Purchase Order:

123

RTI Project Number:

32

Next Exit

Figure 1. Tracing vendor and RTI information form

Next, projects select the file type of the file containing project data, (e.g., ASCII fixed format, comma delimited, Excel, SAS), and the complete file name.

Select Tracing File

Tracing - Provide File Information

Specify the input file format:

☐ ASCII

☐ Comma delimited

☐ Excel

☒ SAS

Select the path to the input file:

Share: j: [\\tints27\\vcd]

Path:

- J:\
- CENTER
- Tracing
- vb

File:

- allforms
- Build_InputFiles_BatchTracing1.log
- Build_InputFiles_BatchTracing1.lst
- Form1.frm
- forms
- FOXUSER.DBF
- FOXUSER.FPT
- Frm_chk.frm
- Frm_chk.frx
- frm_CorrectNames.frm
- frm_CorrectNames.frx
- frm_InputFile.frm
- frm_Main.frm

Main Menu Next Exit

Figure 2. File information form

The list of data items below in Figure 3. is a compilation of data items from all the batch tracing vendors RTI has contracts with. The system uses default variable names but projects can change the variable names as needed. Projects select data items available in the file containing project data and enter variable names if they are different from the default variable names.

Provide Correct Variable Names

Tracing - Provide Variable Information

If your variable name is different than the one listed below, click the corresponding check box and fill in the appeared text box with the correct variable name.

☐	CaseID	
☒	FirstName	FName
☐	MiddleName	
☐	LastName	
☐	Suffix	
☒	Address1	Address
☐	Address2	
☐	City	
☒	State	St
☐	Zip	
☐	Phone	
☒	SSN	SSN1

Back Next Exit

Figure 3. Variable information form

The Visual Basic 6.0 code shown below uses information entered on this and previous forms to construct a parameter string that is used by SAS to select SAS macros to read, standardize, and reformat data in the input file.

```
sCorrectName = Trim(txtCaseID.Text) & "," & Trim(txtFName.Text) & "," & Trim(txtMName.Text) & "," &  
Trim(txtLName.Text) & "," & Trim(txtSuffix.Text) & "," & Trim(txtAddress1.Text) & "," &  
Trim(txtAddress2.Text) & "," & Trim(txtCity.Text) & "," & Trim(txtST.Text) & "," & Trim(txtZip.Text) & "," &  
Trim(txtPhone.Text) & "," & Trim(txtSSN.Text)  
gParameterString = Chr(34) & gParameterString & "," & sCorrectName & Chr(34)
```

The final step in the process is running the SAS application to read, standardize, and reformat data in the input file to conform to the requirements of the batch tracing vendor and the type of tracing selected on the first form. Note that the name of the file containing the SAS code is shown for reference in Figure 4.

Projects select the function they want to perform: run the SAS code, review or print the SAS log file that resulted from running the SAS code, review the SAS list file that resulted from running the SAS code.

Run Program to Generate File for Batch Tracing

Tracing - Generate File for Batch Tracing

Name of the batch tracing program:

SAS Program selected: Build_InputFiles_BatchTracing1.sas

☒ Run batch tracing program

☐ View Build_InputFiles_BatchTracing1.log File

☐ View Build_InputFiles_BatchTracing1.lst File

Print the *.log

Back Submit Exit

Figure 4. SAS processing form

The following Visual Basic 6.0 code starts SAS and passes parameters to SAS via **gParameterString**.

```
IReturnValue = Shell("C:\Program Files\sas\SAS 9.1\sas.exe -sysin " & path_name & "sas_code" & "\" &
FileName & " -sysparm " & gParameterString, vbMaximizedFocus)
```

SAS Application

The SAS application obtains and processes information from the Visual Basic 6.0 application. It then reads project data and generates a file to be submitted to the batch tracing vendor based on information obtained in the Visual Basic 6.0 application.

In the following step, SAS parses the parameter string passed from the Visual Basic 6.0 application and creates a series of macro variables.

```
data _null_;
    delim=",";
    params=tranwrd("&sysparm",",",",",",',''," ");

    call symput("Type",scan(params,2,delim));
    call symput("FileName",scan(params,3,delim));
    call symput("Project",scan(params,5,delim));
    ...
    call symput("CaseID",scan(params,8,delim));
    call symput("FName",scan(params,9,delim));
    call symput("MName",scan(params,10,delim));
    call symput("LName",scan(params,11,delim));
    call symput("Addr1",scan(params,13,delim));
    ...
run;
```

The macro InData uses the file type specified previously to read the data.

```
%macro InData;
%if &Type.=ASCII %then %do;
```

ASCII uses a pre-defined, fixed format record layout and default variable names.

```
%let FName=FirstName;
%let LName=LastName;
%let MName=MiddleName;
%let Addr1=Addr1;
...

DATA DSNAME;
    INFILE &InputTxtFile. LRECL=185 MISSEVER END=EOF;
    INPUT
        @001 &FName. $15.
        @017 &MName. $10.
        @028 &LName. $24.
        @053 &Addr1. $48.
        ...
    ;
run;
```

```

%end;
%else
%if &Type.=EXCEL %then %do;

```

EXCEL uses default variable names and the default worksheet name Sheet1.

```

PROC IMPORT OUT= DSNAME
DATAFILE= "&FileName."
DBMS=EXCEL REPLACE;
GETNAMES=YES;
MIXED=YES;
SCANTEXT=YES;
USEDATE=YES;
SCANTIME=YES;
run;
%end;
%else
%if &Type.=Comma %then %do;

```

For comma delimited input files, project staff can specify variable names on the variable information form.

```

PROC IMPORT OUT= DSNAME
DATAFILE= "&FileName."
DBMS=CSV REPLACE;
GETNAMES=YES;
DATAROW=2;
RUN;
%end;
%else
%if &Type.=SAS %then %do;

```

For SAS input files, project staff can specify variable names on the variable information form.

```

%let sasfile=%scan(&FileName., -2, '\. ');
%let x=%index(&FileName., &SASFile.);
%let inpath=%substr(&FileName., 1, &x-1);

%put &SASFile. &InPath. &x;

libname inlib "&InPath.";

DATA DSNAME;
    set inlib.&SASFile.;

run;
%end;

%mend InData;

%InData;

```

Depending on the batch tracing vendor selected in the Visual Basic 6.0 application, one the following macros is called to generate the file to be submitted to the batch tracing vendor.

```
%macro NCOA(InputFilename);
```

NCOA (LortonData) requires a fixed-format text file.

```
DATA _NULL_;
SET &InputFilename.;

FILE "&Path.\NCOA_Input.TXT" LRECL=170;
PUT
    @001 &FName. $27. /*First Name of Subject to be Traced*/
    @030 &MName. $10. /*Middle Name */
    @040 &LName. $24. /*Last Name */
    @069 &Addr1. $55.; /*Address of Subject to be Traced*/
    ...
;
RUN;
%mend NCOA;
```

```
%macro ACCURINT(InputFilename);
```

ACCURINT requires a comma delimited file and specific variable names. The SAS application renames the variables to meet ACCURINT requirements.

```
DATA ACCURINT_INPUT;
LENGTH
    &FName. $25
    &MName. $10
    &LName. $25
    &Addr1. $60
    ...
;

SET &InputFilename.;
keep
    &FName.
    &MName.
    &LName.
    &Addr1.
    ...
;

RENAME
    &FName. = FirstName
    &MName. = MiddleName
    &LName. = LastName
    &Addr1. = Address
    ...
;
RUN;
```

```

PROC EXPORT
DATA=ACCURINT_INPUT
OUTFILE="&Path.\ACC_&SYSDATE9.CSV" REPLACE
DBMS=DLM ;
DELIMITER=', ';
RUN;
%mend ACCURINT;

```

```

%macro TELEMATCH(InputFilename, ProjName);

```

TELEMATCH requires a fixed-format text file.

```

DATA _NULL_;
SET &InputFilename.;

FILE "&Path.\TM_&SYSDATE9.TXT" LRECL=319;
PUT
    @012 &FName. $50.
    @062 &MName. $50.
    @112 &LName. $50.
    @162 &Addr1. $50.
    ...
    ;
RUN;
%mend TELEMATCH;

```

```

%macro TRANSUNION(InputFilename);

```

TRANSUNION requires a fixed-format text file.

```

DATA _NULL_;
SET &InputFilename.;

FILE "&Path.\TU_&SYSDATE9.TXT" LRECL=183;
PUT
    @029 &FName. $15.
    @059 &LName. $25.
    @087 &Addr1. $50.
    ...
    ;
RUN;
%mend TRANSUNION;

```

```

%macro FIRSTDATA(InputFilename, ACNT);

```

FIRSTDATA requires a fixed-format text file.

```

DATA _NULL_;
SET &InputFilename.;

```

```

FILE "&Path.\&ACNT._&SYSDATE9.TXT" LRECL=247;
PUT
    @001 &FName. $15.
    @016 &LName. $25.
    @049 &Addr1. $48.
    ...
    ;
RUN;
%mend FIRSTDATA;

```

The macro Process_Input generates the file to be submitted to the batch tracing vendor based on information entered in the Visual Basic 6.0 application.

```

%macro Process_Input;

    %IF &VndrID. = LortonData %THEN %DO;
        %NCOA(DSNAME);
    %END;
    %ELSE %IF &VndrID. = Accurint %THEN %DO;
        %ACCURINT(DSNAME);
    %END;
    %ELSE %IF &VndrID. = TeleMatch %THEN %DO;
        %TELEMATCH(DSNAME, &Project.);
    %END;
    %ELSE %IF &VndrID. = TransUnion %THEN %DO;
        %TRANSUNION(DSNAME);
    %END;
    %ELSE %IF &VndrID. = FIRSTDATA %THEN %DO;
        %FIRSTDATA(DSNAME, &Account.);
    %END;

%mend Process_Input;

%Process_Input;

```

CONCLUSION

RTI developed an efficient, user-friendly system to input SAS, Excel, comma delimited, and ASCII fixed format files and to output files in the format required by batch tracing vendors. In addition, SAS log and list files are generated for review by project staff.

CONTACT INFORMATION

Mila Chiflikyan
RTI International
3040 Cornwallis Road
Durham NC 27709
Phone: (919) 316-3360
Email: milachif@rti.org

Jean Richardson
RTI International
3040 Cornwallis Road
Durham NC 27709
Phone: (919) 541-6107
Email: jer@rti.org

Nick Kinsey
RTI International
3040 Cornwallis Road
Durham NC 27709
Phone: (919) 541-6791
Email: nlk@rti.org

Donna Medeiros
RTI International
3040 Cornwallis Road
Durham NC 27709
Phone: (919) 541-8788
Email: djm@rti.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.