Blaise Instrument Extensions with Alien Routers and Manipula

Lilia Filippenko, Chris Carson, Orin Day, and Mai Nguyen, RTI International, United States

1. Abstract

RTI International has conducted many studies involving field data collection on laptop computers. Most of these Computer Assisted Personal Interviews (CAPI) are conducted in respondent's homes and then data is transmitted to RTI. While collecting data for researchers, sensitive personal information is often provided by respondents. This data should be protected on the laptops and during transmission.

There are different ways to protect collected information. This paper describes an easy way to do that by using Blaise instrument with alien routers and procedures in Manipula setup to:

- Enter sensitive data outside of Blaise interview;
- Call external application to electronically obtain proof of respondent consent or other sensitive information;
- Download GPS data from a GPS-tracking device at the end of the interview.

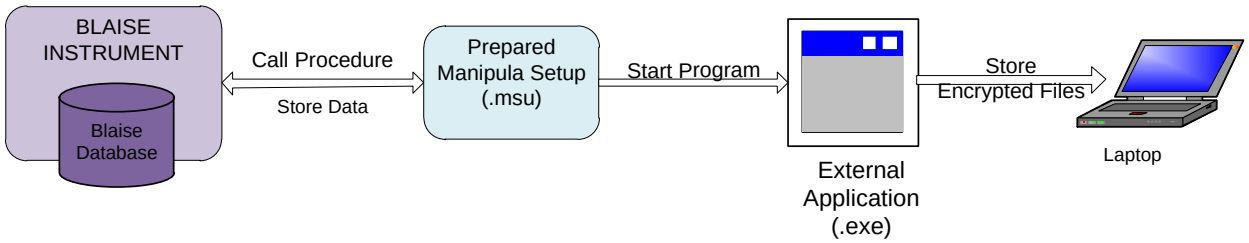
Examples of code in Blaise instrument and Manipula setup are provided in this paper along with description of the external applications used during Blaise interviews.

2. Data Protection on Field Interviewer laptops

The Check Point Full Disk Encryption Software is installed on all RTI field interviewer laptops. So all data, operating systems and temporary files are automatically encrypted by default without relying on the user. Full Disk Encryption performs the encryption transparently to the user, who never needs to bother about what to encrypt and when.

In addition to whole hard drive encryption, files that are ready to be transmitted to RTI are encrypted by the Case Management System. When encrypted zip files with collected data are received at RTI and decrypted, we still want to have some sensitive information encrypted at rest and ready for review only when they are needed. To achieve this goal, such files are encrypted at the very moment when they are created. To satisfy clients' requirements, FIPS 140-2 compliant encryption method is used by the external application called during the interview. Figure 1 shows the process of creating special files during the interview using Manipula setup.

Figure 1. Encryption of Sensitive Data during Interview



We've successfully used all available features of Blaise to call external programs and collect data outside of Blaise. Previously, we used alien routers created as DLL in .NET. However this approach had a drawback during the development stage for clients' testing because alien routers were required to be registered on the client's machines and that was not always permitted.

When a new feature was added to Manipula to use a parameter to set variable meta names at run-time, we found that implementation was very easy and preparation of installation packages for testing and deployment on field interviewer laptops were much simpler. The following section has a detailed explanation of what needs to be done in a Blaise datamodel and a Manipula setup to achieve this.

3. Collecting Data Outside of Blaise Interview

One example of sensitive data requiring immediate encryption after entry is a social security number (SSN). Collecting and storing the SSN separately from the survey data ensures respondents that their data will be available only for limited research and that it will be specially protected before it is delivered to researchers. A description of how this is done is shown below.

3.1 Implementation in Blaise Instrument

In the Blaise datamodel, fields where data will be stored in Blaise database are defined along with an alien router that redirects execution of a procedure in a prepared Manipula setup:

- block BSSN4 with field SSN4;
- router DataEntry (keyword BLOCK is used to apply the same router for another block; name of the meta file is passed as parameter);
- field SSN4 as BSSN4.

Figure 2. Define Fields and Alien Router

```
BLOCK BSSN4 {Used to ented SSN outside Blaise }
    FIELDS
        SSN4 ""/"The last 4 digits of Social Security number": STRING[4]
ENDBLOCK

ROUTER BLOCK.DataEntry
    ALIEN('AHPSRouter.msu /KMyMeta=$dictionaryname', 'DataEntry')

FIELDS
   SSN4
   ENUS
   "We are committed to protecting the confidentiality of all the information you
   give us. What are the last four digits of your Social Security Number?"
    : BSSN4
```

In the RULES section of the Blaise datamodel, a defined router DataEntry is used instead of ASK method to handle the field in the block SSN4. Also, when data is saved outside of Blaise, we need to have information about how the respondent answered the question to monitor collection of SSN data and ask that question only once if data was collected and encrypted. That is achieved by checking the value of SSN4 - “0000” which means that SSN was collected.

Figure 3. Define a Routing Method

```
RULES
SSN4.keep
IF SSN4.SSN4 <> '0000' THEN
    {call Router DataEntry}
    SSN4.DataEntry
    CHECK
        SSN4.SSN4 = NONRESPONSE OR SSN4.SSN4 = '0000'
        "PLEASE RE-ENTER THE LAST 4 DIGITS OF SOCIAL SECURITY NUMBER."
ENDIF
SSN4.keep
```

During the interview an audit trail file with all answers is created to help troubleshoot some problems and to monitor flow of the interview. The audit trail file is an ASCII file that can be opened in any text editor. By collecting the SSN outside of Blaise, the audit trail file is bypassed and security is enhanced by keeping this sensitive information out of the audit trail file.

3.2 Implementation in Manipula Setup

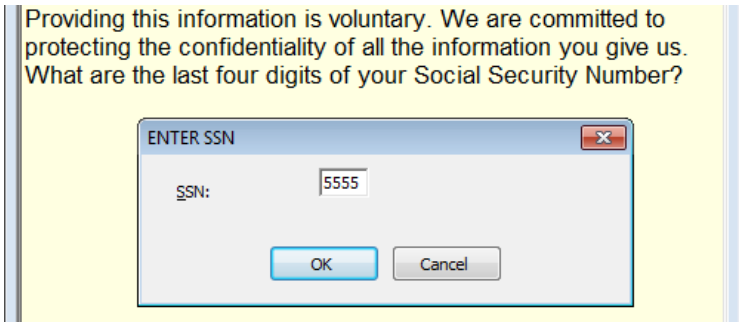
When the Manipula setup AHPSRouter is called during the interview, all information in the current interview is available through "INTERCHANGE =SHARED" setting at the beginning of the Manipula setup. The reserved word "(VAR") is used instead of the meta identifier so that Manipula setup knows that the prepared datamodel that corresponds with the meta identifier is only known at run-time.

Figure 4. Define Meta and Settings in Manipula Setup

```
PROCESS AHPSRouter
SETTINGS
  DATEFORMAT = MMDDYY
  DATESEPARATOR = '/'
USES
  mymeta (VAR)
TEMPORARYFILE mydata:mymeta
SETTINGS
  INTERCHANGE=SHARED
```

A procedure DataEntry and a dialog to collect last four digits of the SSN are defined in the Manipula setup AHPSRouter as shown in Figure 5.

Figure 5. Entering Data in Manipula Setup



When the OK button is pressed and the SSN is validated to contain only digits, an external application developed for a study is called to encrypt it with a specified encryption key.

Figure 6. Abbreviated Version of the Procedure to Collect SSN in Manipula Setup

```
PROCEDURE DataEntry
AUXFIELDS
    CaseID : STRING[8]
INSTRUCTIONS
    IF (ROUTERSTATUS=BLRSPREEDIT) THEN
        FieldName := uppercase(ACTIVEFIELD)
        CaseID := mydata.GETVALUE('main_case.zrid')
        {Show Dialog in DEP}
        SSNDialog4
        {... more code to validate data entered ...}
        IF BUTTONs = Okay THEN
            {Call external program and pass case ID and SSN for encryption}
            Reslt:= RUN('AHPStools.exe Crypto' + ' ' + CaseID + ' ' + UserSSN4)
            IF Reslt = 0 THEN
            {File with encrypted data is created - save in Blaise with zeros}
                IF (FieldName = 'SSN9.SSN9') THEN
                    mydata.PUTVALUE('SSN9.SSN9', '000000000')
                ELSEIF (FieldName = 'SSN4.SSN4') THEN
                    mydata.PUTVALUE('SSN4.SSN4', '0000')
                ENDIF
                {Move to a next field in the interview}
                SETALIENROUTERACTION(BLRANEXTQUESTION)
            ELSE
                {... more code ...}
            ENDIF
        ELSE {return to the same field }
            SETALIENROUTERACTION(BLRAEDITQUESTION)
        ENDIF
    ENDIF
ENDPROCEDURE
```

The application saves the encrypted value in a text file with a name that has the primary key, Case ID, as part of it. This will allow RTI to decrypt data from the file later, when it is ready for researchers to use. The file is saved on a laptop in a folder from which it will be transmitted along with other collected data for the case.

4. Obtaining Respondent Signatures with RTI's DocMan

Another example of collecting sensitive information is obtaining respondent signatures on different forms. In an effort to protect the confidentiality of survey participants and to ensure the security of all data that field interviewers use and collect while working cases, RTI has created an electronic document management system named DocMan.

4.1 DocMan Description

DocMan is the collective name for a set of systems developed by RTI to eliminate paper documents from field data collection. DocMan provides field interviewers a facility for electronic collection of signed forms or documents, via legal electronic signatures using a USB signature pad or through scanning paper forms using a USB portable scanner. Such forms could include informed consent, incentive receipts, and so on. Once collected and verified by the interviewer, DocMan encrypts the forms and provides secure transfer of those forms back to the RTI home office. DocMan also provides secure transfer of read-only documents, such as locating information, as well as the capability to key or scan additional information regarding the case. DocMan was first used in 2007 and has been used on many RTI field studies.

Notable components of DocMan include:

- DocMan Executable – .NET application,
- DocMan Information Files – set of PDF files,
- DocMan Forms – set of signed or scanned documents,
- DocMan Case Notes – notes in electronic format used to replace paper case folders.

DocMan Executable resides on the field laptop that the interviewer uses as an interface to access three primary features:

1) To view DocMan Information Files,
2) To collect signed forms as Forms, and
3) To record DocMan Case Notes.

DocMan can be deployed with any or all of the features above. The effectiveness of DocMan was enhanced by enabling it to be called directly from Blaise at the correct point in the interview. An example of how DocMan is used with Blaise is provided in Section 4.2.

DocMan Information Files – PDF files with contents and format specified by the project that are included in case preloads to provide case information to the field interviewer beyond what is available in the Case Management System. PDF files are encrypted, password secured, and can include security features to make them unprintable.

DocMan Forms – described above, these are signed or scanned documents digitally acquired and encrypted, signed via electronic signature pad (ePad) or scanned with a portable scanner (equipment examples below).

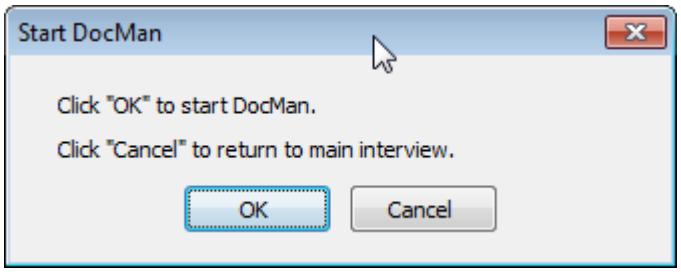
Figure 7. Equipment used with RTI DocMan: Interlink ePad (left) and Brother DS-620 Mobile Color Page Scanner (right)

DocMan Case Notes – The option to collect interviewer notes on the laptop and then securely transmit those notes and event/status/comment information to RTI. Once received the information is decrypted and periodically collated and compiled into secure PDF documents. Those documents are released to RTI staff and transmitted to field supervisors. Implementing this system involves configuring the laptop application and Case Management System, and customizing the format of the Case Notes file to match project needs. For longitudinal studies the Case Notes from past rounds of data collection are included as DocMan Information Files, securely replacing paper case folders.

4.2 Blaise and DocMan

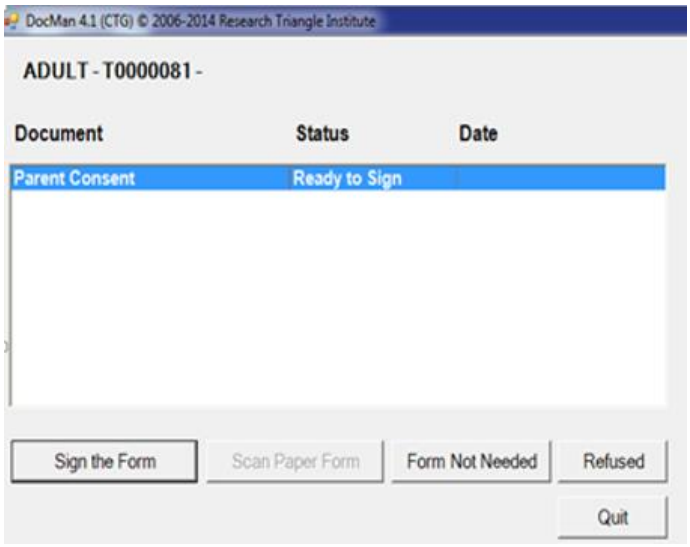
DocMan is used during the Blaise interview to electronically obtain proof of respondent consent to the interview or other information. The Blaise implementation of a call to DocMan is almost the same as described above in section 3. When an interviewer reaches a field where DocMan is called, a dialog is defined in a Manipula setup to confirm that the field interviewer has everything ready to start collection of a signature or to scan a document. While our example below includes the process for signing a document with the ePad, the process flow for scanning a document is similar.

Figure 8. Call to DocMan from the Blaise Interview



Clicking "OK" will automatically launch DocMan and display a list of usable forms for this portion of the interview. DocMan will automatically present the field interviewer with the appropriate options. Figure 9 shows the window displayed in DocMan during the informed consent portion of an interview. The available document has a status of "Ready to Sign" meaning the document had not yet been completed.

Figure 9. DocMan Selection Screen and Document Status



Clicking "Sign the Form" will open the DocMan template selection screen where the field interviewer will be asked to select from prepared templates that can be used in the current situation. The Word document is then ready for the respondent's electronic signature. The field interviewer will ask the respondent to review the form on the computer, and will then hand the electronic signature pad (ePad) to the respondent for her/his signature.

After all required signatures have been collected and the field interviewer confirms that form was completed correctly, DocMan recognizes that the document has been closed and returns back to "DocMan Selection Screen and Document Status". The status of the document is then changed from "Ready to Sign" to "Complete – e-Pad". At this point the field interviewer will press "Quit" to return to the Blaise interview.

Control is then passed to the Manipula setup where the completion status of the signed form is determined by checking existence of the encrypted file created by DocMan in a specified folder. Fields defined in the datamodel to save the status of the document (completed or not) and some timing data to monitor field interviewer performance are calculated in a Manipula setup. An example is shown in Figure 10.

Figure 10. Run DocMan from Manipula Setup

```
{Create name of encrypted file - PDF or Word document}
IF (DocID = 'S1') OR (DocID = 'S2') THEN
    strFileName := strPathName + DocID + '\' + CaseID + '_' + DocID + 'E.pdf'
ELSE
    strFileName := strPathName + DocID + '\' + CaseID + '_' + DocID + '.docx'
ENDIF
{Start DocMan}
strRun := '"c:\Program Files\DocMan\DocMan.exe"' + ' /CASEID=' + CaseID
          + ' /' + strTrProd + ' /LAUNCHDOC=' + DocID + ' /INSTLANG=B'
Reslt:= RUN(strRun, wait)
IF (Reslt <> 0) AND (Reslt <> EMPTY) THEN
    DISPLAY('Problems with starting DocMan. Please report to FS.', WAIT)
ELSEIF Reslt = 0 THEN
    {Check that file was created and set completion status}
    IF FILEEXISTS(strFileName) THEN
        Status := '491'
    ELSE
        Status := '309'
    ENDIF
    mydata.PUTVALUE(BlockName + '.Status', Status)
    {... more code to save data in Blaise ...}
    {Move to a next field in the interview}
    SETALIENROUTERACTION(BLRANEXTQUESTION)
ENDIF
```

Protecting subjects' confidentiality and maximizing data security are increasingly critical objectives of field research. DocMan eliminates the need to collect and return paper documents from the field while sufficiently collecting electronic documents during an interview by using an alien router in Blaise interview with a Manipula setup. This approach allows the field interviewers to do their job more efficiently.

5. Download GPS Data on Laptop

Global Positioning System (GPS) is used by many people every day. One of our recent study clients requires the use of a GPS unit to capture geo-coordinates of the place where interview is being conducted. Collected data is supposed to be entered into a special Blaise instrument. We decided that having geo-coordinates will also provide us with an additional way to verify the field interviewers' work. Following sections describe how collection of geo-coordinates is implemented in the field and two ways to use collected data for verification.

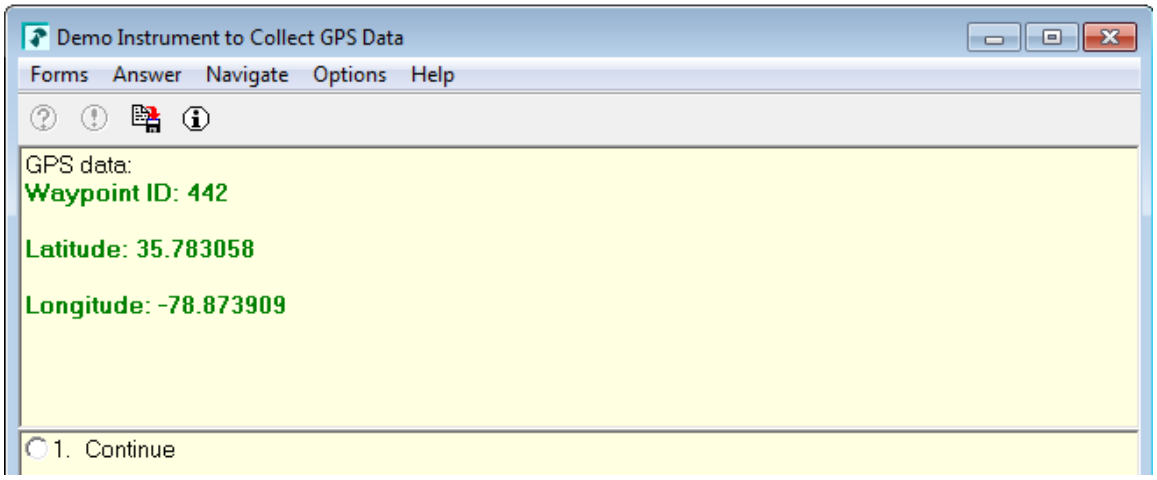
5.1 Field Implementation

We selected the GlobalSat DG-100, a GPS data logger that records track data, because of its simplicity and low cost (approximately $22 each). The DG-100 records time, date, speed, altitude and GPS location at preset intervals. It comes with a USB plug to download readings to any USB capable laptop or desktop and with Windows based software utility. There is no LED display. Field interviewers are simply instructed to turn it on, wait for a few minutes until the green light flashes, and then press the silver button.

Figure 11. GlobalSat DG -100

After the coordinates have been captured, the field interviewers launch another interview called the Field Interviewer Observations Module where they are asked to plug the DG-100 into their laptop. After it is confirmed that the device is plugged in, the Manipula setup is called to execute an external application "DataLogger" to download data from the GPS device and pass geo-coordinates to the Blaise database. The result of this process is shown in Figure 12. Implementation in the Blaise interview and the Manipula setup are the same as described above in 3.1 and 3.2.

Figure 12. Data downloaded from DG-100 in Blaise Database



5.2 Data Logger Application

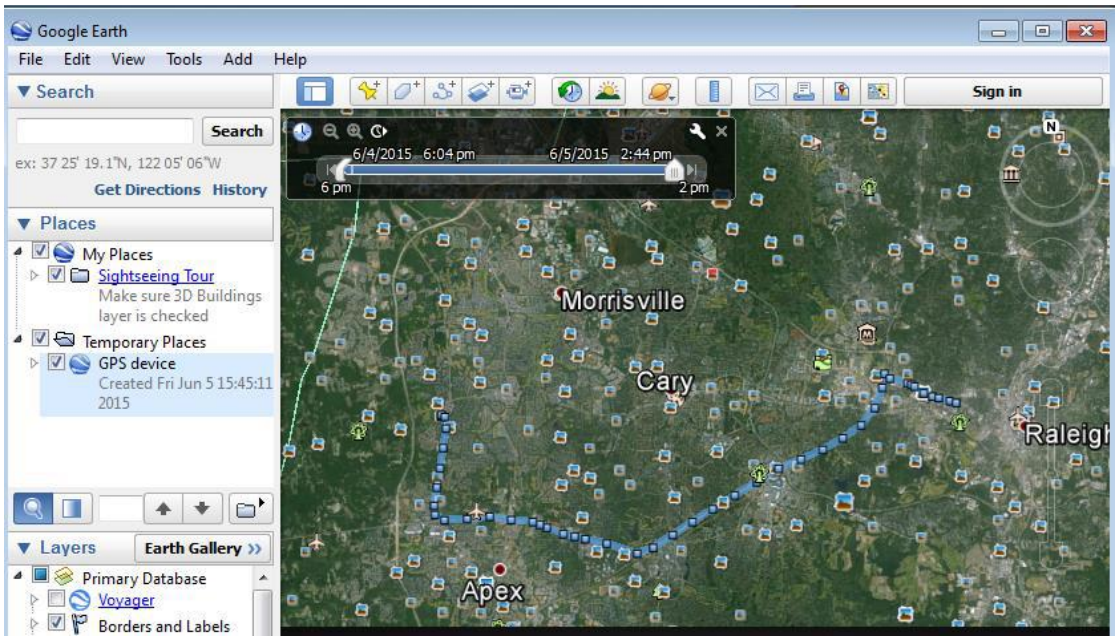
For the field interviewers, the process of downloading GPS readings looks seamless. But Data Logger application developed in .NET does a lot of work:

- it identifies the COM port to which DG-100 is connected on laptop,
- it checks that DG-100 is ready to upload data,
- it executes application gpsbabel.exe to save data on laptop in two files - <CaseId>.csv and <CaseId>.kml.
- it reads a .csv file to find geo-coordinates with timestamp as close as possible to the CAPI interview timestamp,
- it creates a temporary file with data that is used by Manipula setup to save geo-coordinates in the Blaise database.

Although DG-100 comes with its own utility to download data, it can only work in interactive-mode. So we decided to use free software called GPSBabel (www.gpsbabel.org.) in a batch mode.

Both files saved on laptop are also transmitted to RTI along with the data from the Field Interviewer Observations Module. Excel can be used to review the .csv file and the .kml file can be viewed in Google Earth as seen in Figure 13 below. In this .kml file, the DG-100 was left running and the whole route captured.

Figure 13. kml File viewed in Google Earth



5.3 Field Interview Verification

GPS capture is one of the most recent advancements in field interview verification. Verifying that the Field Interviewer was present at the respondent's address at the date and time the interview took place is a very good indication that the interviewer actually conducted the interview with the respondent. However, it is not an absolute certainty. Other complimentary methods of interview verification should also be used. These include an analysis of timing data and listening to recorded segments of the interview (CARI).

5.3.1 SAS® Proc Geocode

The street-level geocoding to an address is done using SAS® Proc Geocode. This procedure can be used to look up physical street addresses and return their corresponding latitude and longitude coordinates. These coordinates can then be compared to the coordinates captured on the DG-100 and some simple calculations can then be employed to compute the distance between the two points. Reports in SAS are created every night for the project staff to review. Ideally, these comparisons will show a very small distance between the captured coordinates and the physical address. When this does not happen, the case can be prioritized for further scrutiny through the other methods used for interview verification.

There are several good examples on the SAS website how to use SAS® Proc Geocode. The default database SAS/GRAPH uses for street level geocoding is SASHELP.GEOEXP and includes addresses from Wake County in North Carolina, where the SAS headquarters is located. To get the data needed

to geocode throughout the United States, SAS provides a free download from its SAS Maps Online website:
http://support.sas.com/documentation/cdl/en/graphref/65389/HTML/default/viewer.htm#n02y3yabtlqatsn16gp2fo51yo7p.htm#n1nmfmy3wf1kc8n0zw565lb41mjt

5.3.2 Google Maps API / Google Geolocation API

As an alternative to SAS® Proc Geocode the Google Maps API can be used. The Google Maps API provides multiple comprehensive services, including Mapping, Directions, Places and Geolocation. The APIs support desktop and mobile devices, and in all popular programming languages: Java, C#, JavaScript, Python, etc. Our code samples will be focused on the Geocoding service using Python. An excellent introduction to the Python language can be found on this website:
https://www.python.org/about/gettingstarted/

Figure 14. Google Geolocation API example

```
from pygeocoder import Geocoder

address = '1600 Pennsylvania Ave., Wanshington, DC'

# Geocoding
result = Geocoder.geocode(address)

# output results
print(result.formatted_address)
print(result.coordinates)
```

```
1600 Pennsylvania Ave SE, Washington, DC 20003, USA
(38.8791981, -76.9818437)
```

The Google Geolocation API also includes reverse geocoding (address lookup) functionality that can be used for looking up address from a geo-coordinate. The Google Geolocation API is straightforward to use. A unique functionality supported by the Google API that is not available in the SAS procedure is the ability to do reverse geocoding. Further documentation about Google Maps API is available at this website:
https://developers.google.com/maps/web-services/overview

6. Conclusion

External applications have become a common requirement on many studies and Blaise has more than one way to implement them. However, we find that the use of an alien router and a Manipula setup that redirects execution is the simplest and most efficient solution: it reduces development time and is easier to debug. RTI has used this approach extensively for projects that need new innovations to accomplish their goals and to accommodate increased levels of data security. We hope that this option continues to be available in Blaise 5 for future studies.