

Using the Blaise® Alien Router For Audio-Recording of In-Person Interviews

M. Rita Thissen, Sridevi Sattaluri and Lilia Filippenko, RTI International

1. Introduction

The Blaise alien router is a useful technique for managing calls to external software. In this paper, we describe use of an alien router for activation of audio-recording software for in-person surveys.

During in-person field interviews, the performance of field staff and the effectiveness of item wording are notoriously hard to monitor. Traditional methods of live observation during the interview can bias results through the presence of the observer, and verification call-backs to the respondent to confirm authenticity and professional behavior may not provide enough detail to assure high quality data collection. For computer-assisted personal interviewing (CAPI), digital audio recording offers an efficient alternative which allows evaluation of interviewer and item performance during data collection (Thissen et al, 2007).

We describe a practical technique in which Blaise (Statistics Netherlands) instruments can be enabled for computer audio-recorded interviewing (CARI). This approach comprises an external file for specifying items to be recorded, an alien router which activates the laptop's sound-recording system, and block-level activation of the router within a Blaise instrument. Ideally, a system for audio recording should require little programming, execute undetectably, provide a flexible choice of which items to record, allow respondent consent to be given or revoked at any time and store recordings in clearly identified files. This system offers all of those advantages. By invoking the router on specific blocks or on the entire instrument, minor programming effort can turn a standard implementation into a CARI-enabled questionnaire.

2. The Blaise Alien Router

Blaise software provides ways to call external procedures, including a technique termed the “alien router.” Before and after each question is asked on-screen, a process known as the router executes instructions regarding layout, checks and other operations which are not apparent to the interviewer. Custom actions can be added to this process, by creating an alien router which runs in addition to (or instead of) the native router to “ask” the question. The alien router can communicate both with the instrument and with external software, such as executables or Windows components. The Blaise Component Pack (BCP) is required for execution of an alien router. ActiveX has been supported since BCP version 2 and Blaise 4.6. Additional information and examples are provided as part of the documentation in the Blaise Online Assistant.

3. Computer Audio-Recorded Interviewing (CARI)

Survey researchers have discovered value in computer audio-recorded interviewing (CARI) (Biemer et al, 2000). This approach helps improve the quality of data collection through unobtrusive recording of the audio portion of in-person interviews (Thissen and Rodriguez, 2004) or telephone interviews (Suresh, 2005); much as silent monitoring has

been used to ensure quality at many call centers. With CARI technology and consent from the respondent, sound files are recorded during the interview, transmitted to a central site and monitored for proper performance of the survey protocol.

Although there are several advantages offered by implementing CARI, perhaps the most compelling reason is to confirm the authenticity of data at lower cost than traditional verification methods. CARI can act both as a deterrent to curbstoning and as a tool for detecting questionable interviews. Informal feedback suggests that interviewers who are aware that monitors may listen to parts of each interview are less likely to falsify data, because the audio file acts as a “witness” to their actions. In this way, the simple presence of CARI can reduce cheating.

While evaluating authenticity, monitors may also rate interviewer performance for feedback. Performance characteristics may be positive or negative, resulting in praise or prompting additional training. Such feedback is especially important in the first few weeks of a novice interviewer’s work, at the start of a new survey, or when conditions for obtaining the interviews are known to be challenging.

CARI can be used to evaluate the usability of questionnaire items. The audio recording of an interviewer’s presentation of an item and the subject’s response provides a clear indication of whether the item is phrased in a way as to be read easily by the interviewer and understood easily by the respondent. Survey items which evoke negative reactions or require frequent explanation are detrimental to the response rate and increase the level of burden on both interviewer and subject. Using CARI, especially during field testing of an instrument, allows the survey specialist to evaluate the effectiveness of the questions.

A complete CARI process includes several steps. Recordings are made in Windows wave format and later compressed to mp3 format to reduce the size before transmission. Automated case management processes transmit the files, record their arrival in a database and present them for review through an interactive interface called the CARI Monitoring system.

1. Recording during the interview
2. Compression and encryption (optional)
3. Transmission and storage
4. File tracking and management
5. Monitoring and evaluation

This article discusses the first step of the process, collecting the audio recordings as part of questionnaire instrumentation.

4. CARI System

Audio recording can be added to Blaise instruments by using either of two programming approaches. One approach uses a Blaise procedure which in turn invokes an external application to start and stop the recorder. Using this approach requires custom programming within Blaise in every place the recording application needs to be invoked, with fields that keep track of whether recording is already in progress or needs to be started or stopped (Thissen and Rodriguez, IBUC 2004). The second approach is described here, using an external item list and an alien router instead of custom rules and

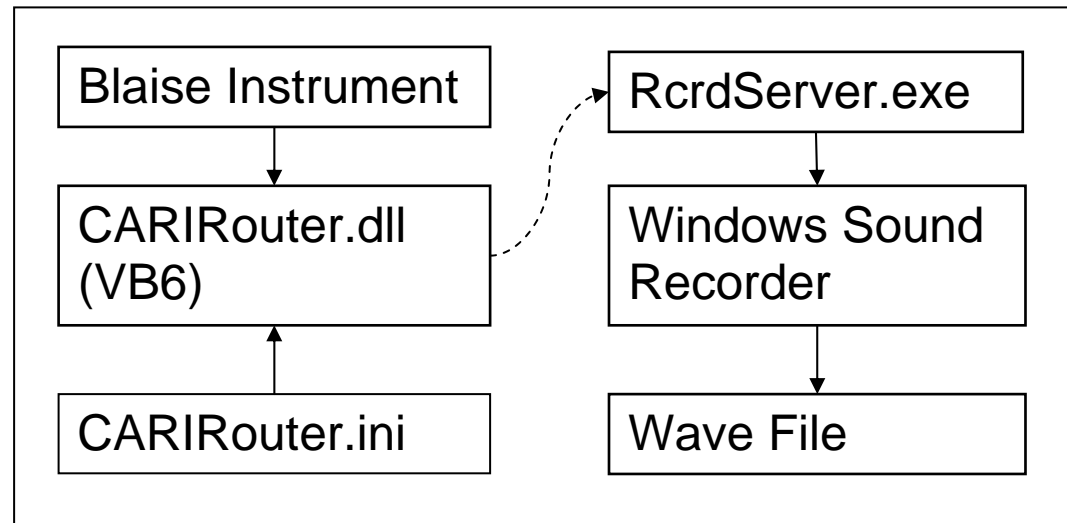
an alien procedure. In the second approach, instrument modifications for CARI are much simpler once the router has been implemented, because the same router may be reused for multiple surveys.

As with any programming challenge, there are several ways to meet the requirements. At RTI, the CARI alien router implementation consists of 5 parts:

- CARI Alien Router (CARIRouter.dll)
- RcrdServer.exe – a standalone executable that starts and stops recording by communicating with the Windows Sound Recorder.
- Blaise instrument – with block-level activation of the alien router
- CARIRouter.ini – an external text file that lists the items to be recorded

By using the native Windows Sound Recorder, performance is optimized during use and drag on processing is minimized. This approach is illustrated in Figure 1.

Figure 1. RTI's approach, using an alien router for CARI



4.1. The CARI Alien Router and RcrdServer

Blaise alien router technology allows the invocation of an external component before every survey item is given focus and after it loses focus. Use of the alien router externalizes the complexities of tracking the recorder state and offers the option of maintaining a text list of items to be recorded, external to the instrument. This reduces the complexity of instrument programming and allows easy modification of the list of items to be recorded, without any need to modify the data model or recompile the instrument (Thissen and Sattaluri, 2006).

Our goals were to implement an alien router which would do the following:

- Successfully start and stop the sound recorder from Blaise
- Successfully create audio files with the specified items
- Provide no indication onscreen that recording was starting or stopping.

- Decrease the programming effort as compared to using an alien procedure
- Provide a flexible method for CARI item selection

In our implementation, CARIRouter.dll is an ActiveX component, implemented in Visual Basic 6.0, which combines information passed in from the Blaise instrument about the current item and the CARIRouter.ini file to determine when to start and stop the recorder. It sends the start and stop commands to the RcrdServer.exe which in turn activates the Windows Sound Recorder. The router contains two functions (ManageCARI, StartCARI.), a reference to the Blaise API, and a form for communication with the sound recording system. Communication with RcrdServer takes place through Windows Sockets. As an alternative, the router could be programmed to directly control Windows Sound Recorder.

Figure 2. The ManageCARI function in CARIRouter.dll.

```
Public Sub ManageCARI (DB As BIAP14A2.Database, DS As BIAP14A2.DepState)

On Error GoTo Err_ManageCARI
strBlaiseDBPath = DB.DataFileName
strBlaiseDBPath = Mid(strBlaiseDBPath, 1, InStrRev(strBlaiseDBPath, "\") - 1)

InitializeRouter strBlaiseDBPath

consentflag = (DB.Field("CARIConsent").Text = "YES")
CaseID = DB.KeyValue
If consentflag Then
    'Get hold of the currently active question
    'The properties of the DepState parameter need to be used for this
    Set Layout = DB.Screens.LayoutSetCollection(DS.LayoutSetIndex)
    Set Parallel = Layout.ParallelCollection(DS.ParallelIndex)
    Set Page = Parallel.StoredPageCollection(DS.StoredPageIndex)
    Set Quest = Page.QuestionCollection(DS.QuestionIndex)
    Dim QID As String
    QID = Quest.Field.Name
    ...
    If DS.AlienRouterStatus = blrsPreEdit Then
        If bRecordEntireSurvey Then
            If Not CARIChecked Then
                StartCARI CaseID, QID, trig.Timer
            End If
        Else
            ...
        End If
    ...
End If
...
End If
...
End sub
```

The Blaise Instrument calls a function, "ManageCARI," within CARIRouter.dll before and after the presentation of each item. The CARIRouter loads up the list of items from the CARIRouter.ini file when it is called for the first time. During every call to the CARIRouter, the Blaise instrument provides information to the router as to which item is

next on the path and if execution is before or after the presentation of the question item. The CARIRouter then matches the item name against its list of items to determine if it needs to be recorded or not. The router then communicates with the RcrdServer.exe to either start or stop the recording.

For the alien router to work in an environment where multiple Blaise databases may be present, the ManageCARI function has been defined to accept the Blaise database name and DepState objects as parameters. The code sample in Figure 2 shows part of the ManageCARI function to demonstrate the interaction between the Alien router and the Blaise Instrument.

The DB.FileName property holds the location of the Blaise database, which is used in turn to determine the location of the INI file. The InitializeRouter function reads in the values from the CARIRouter.ini file and initializes the connection to the RcrdServer.exe. Code inside the InitializeRouter function makes sure that the initialization happens only once. Examining the code in Figure 2 reveals several important actions:

- The parameters DB and DS are passed in by the Blaise Instrument and they identify the Blaise database and its current status.
- The DB.Field method can be used to retrieve the current field value of any Blaise database field.
- The Page.QuestionCollection(DS.QuestionIndex) provides the reference to the current question in the instrument
- The DS.AlienRouterStatus provides the information about whether the function was called before or after the presentation of the question item.

Figure 3 shows code from the StartCARI function. This function uses a flag, bOverwriteWavfiles, that is initialized from the CARIRouter.ini to determine if previous recordings with the same question name will be overwritten or not. If the flag is set to “yes” within the ini file, the contents of the recording file will be overwritten each time the question is asked. This may be desirable, in a situation where each a question is asked multiple times and only the final response should be kept. On the other hand, it may be desirable to set the flag to “no” if the respondent is allowed to back up through an item and pass through it again, because on the later passage the question may only be displayed briefly in transit, without any response. This sequence of actions may happen when a respondent or interviewer backs up through several questions to view the answer of an earlier question, and then uses arrow keys to move forward to the last unanswered question.

One of the alien router’s actions is to signal the recording software to start or stop, by communication with RcrdServer. Signaling is handled through a function SendRequest in the form frmRcrdClient. The first parameter of SendRequest determines that the recording needs to be started. The second parameter defines the timer value if any. The third is the name of the wav file to create. At RTI, we use a long file name, created with case ID and item name as part of the file name, to allow unambiguous identification.

Because the user may navigate back and forward through the instrument, possibly moving from items designated for recording to blocks or questions that should not be recorded, logic is required to keep track of whether a recording is already in progress. Similarly, if the respondent changes their consent from “yes” to “no,” the software must

be aware of the current state of recording. The `mbCARISStarted` and `CurrentTimer` variables are used to keep the status of whether a recording is currently in progress, to allow it to be stopped or started appropriately. The `StartCARI` function is used to start the recording, and the `StopCARI` function is used to stop it.

Figure 3. The `StartCARI` function in `CARIRouter.dll`.

```
Private Function StartCARI(CaseID As String, QID As String, Optional Timer As Integer = 0) As Long
    Dim s As String
    If bOverwriteWavfiles Then
        mStrSndFileName = CaseID & QID
    Else
        mStrSndFileName = CaseID & QID & Format(Now(), "mmddyyyyhhnnss")
    End If
    frmRcrdClient.SendRequest "1," & Timer & "," & mStrSndFileName
    mbCARISStarted = True
    If Timer > 0 Then
        CurrentTimer = Timer
        StartTime = Now()
    End If
End Function
```

Within the alien router, `frmRcrdClient` communicates with `RcrdServer` by using the WinSock ActiveX control. It initializes the WinSock control to use a particular port, to which the `RcrdServer` listens. The `SendRequest` function sends the request using the `SendData` method of the WinSock control. Requests to begin recording include a 1 to start recording or a 0 to stop recording. If the recording is to be made for a specific length of time, that timer value is also passed along. Finally, the router passes the file name to be used in saving the audio file.

For use with the Blaise instrument, `CARIRouter.dll` and `BlAPI4A2.dll` must be registered on the Windows laptop. Their location may be in the `System32` folder or elsewhere on the computer. In our application, `Microsoft WinSock.dll` must also be present and registered.

4.2. Enabling the Instrument for CARI

After implementing an alien router dll, the next step is to reference it and invoke it from the Blaise instrument. Within the instrument or block header, the presence of the alien router is indicated by the `ROUTER` statement:

```
ROUTER RouteRules ALIEN ('CARIRouter.CARISRecorder', 'ManageCARI').
```

The syntax of this statement, which alerts the DEP to the existence and identity of the alien router, can be found in Blaise Online Assistant documentation under the heading “Router.” In our implementation, it provides a name “RouteRules” which acts as the name of the alien router within the Blaise code. `CARIRouter` is the name of the dll itself, `CARISRecorder` is the public routine inside the dll, and `ManageCARI` is the name of the function which is activated.

Add the definition of the router to all blocks where CARI may need to start or stop recording. If CARI might be started in one block and deactivated in another, the router has to be associated with both blocks. The router can be associated with a top-level block, and it will affect all sub-blocks. Only one router can be active at a time, and so if

another alien router or external call has been defined in the instrument, block structure may need to be revised to keep the two isolated. See Section 5 below for additional comments about constraints on alien router usage.

The Blaise code requires some relatively simple changes to incorporate the CARI functionality. Every Block that contains questions that may be recorded should have the ROUTER ALIEN keywords as shown in the highlighted code in the following example.

Figure 4. Associating an alien router with a specific block

```
BLOCK Bperson
PARAMETERS
    {define parameters here...}
ROUTER RouteRules ALIEN ('CARIRouter.CARIRecorder', 'ManageCARI')
FIELDS
    {define Fields here...}
RULES
    {define logic here..}
ENDBLOCK
```

Refer to the Router in the Rules as shown in the highlighted code. The alien router will only be invoked if the router name is added to the block when it is called in the Rules. Figure 5 shows a simplified example of Fields definitions, including a question asking for consent to record the interview. In the subsequent Rules section, the block HHRoster has been enabled for CARI recording by adding “.RouteRules” to the call.

Figure 5. Calling a block with an associated alien router

```
{Locals and other definitions go here...}
FIELDS
    CARIConsent : tyesno
    HHroster : array[1..3] of Bperson
    {more fields go here ....}
    Q2 "Has consent been given?": bconsent

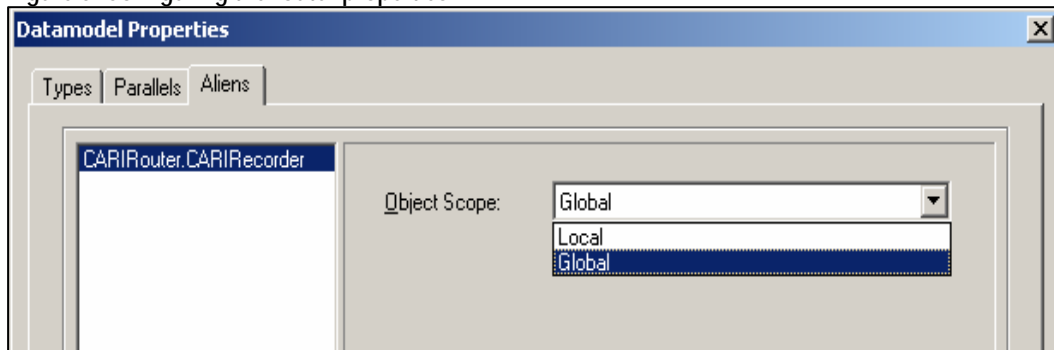
RULES
    {more rules go here...}
    CARIConsent.keep
    Q2
    For I := 1 to 3 do
        HHroster[I].RouteRules
    enddo
    {more rules go here...}
ENDMODEL
```

At RTI, we request consent at the start of the interview to permit recording, and we allow the respondent to change that response at any time during the session. For example, if a respondent initially refuses and later decides it would be acceptable, the CARIConsent item can be changed from “No” to “Yes.” Conversely, if a respondent initially assents, permission can be revoked by changing CARIConsent to a value of “No.” To manage revocable consent, define a Consent block that asks for the respondent consent as a routed parallel block. If multiple consents must be stored, as in a household interview, use a separate kept variable to hold the value of the current respondent’s CARIConsent.

Depending on the rules of the state or country in which the interview is conducted, it may or may not be legal to record the consent question itself.

There is one configuration item to set, for the CARI instrument to run well. Set Object Scope for the alien router in the Data Model Properties to Global so that a single instance of the router runs when the instrument runs, as shown in Figure 6. Otherwise, a new instance of the router will be created for each item, resulting in a drain on performance and memory.

Figure 6. Configuring the router properties



The CARIRouter handles instrument navigation anomalies such as backing up or breaking off while recording is taking place, instrument logic complexities such as looping and skips which may need to start and stop the recorder repeatedly, and as mentioned above the option of revocable consent.

4.3. Specifying Items to be Recorded

At this point, the CARIRouter has been defined and the Blaise instrument has been adapted to make use of it. Block-level association in the Rules invokes the alien router. How does CARIRouter know when to turn the recorder on or off? At RTI, that information is contained in an initialization (ini) file external to both the router and the instrument. There are many possible ways to communicate the CARI item list to the router, and we chose this approach for its flexibility and ease of use.

At RTI, the CARI ini file is a simple external text file for specifying items to be recorded. Our ini scheme offers great flexibility in selecting portions of the interview for recording, including:

- Single items as individual files
- Sequential items in a single file
- Timed recordings
- Starting a new recording when an earlier one is still in progress
- Recording an entire survey into a single file
- Indicating specific elements from an array

Logic to control recording under these circumstances is specified in the ini file and executed by the alien router.

Because it will be parsed by the alien router, the ini file has a defined structure and syntax for the statements contained in it. The file must be named “CARIRouter.ini” and reside on the laptop in the same folder as the Blaise database. An example of a simple ini file is given in Figure 7, and its details are explained in subsequent paragraphs.

Any line in CARIRouter.ini which begins with a semi-colon as the first character or is completely blank will be ignored. This provides a convenient way to include comments or descriptive information. Each section starts with a header line, which is [Options] or [Items].

Figure 7. Contents of a simple CARIRouter.ini file

```
; CARIRouter.ini

[Options]
RecordEntireSurvey = No
OverwriteWavfiles = No

[Items]
; Specify the itemcount as the number of entries that follow.
ItemCount=4
; Record 20 seconds following the start of Q5, regardless of how many items are included
Item1=Q5, CONTINUE, 20 ; timed recording for 20 seconds
; Record items Q12 and Q13 as a single file
Item2=Q12, START
Item3=Q13, STOP
; Begin at the third element of Q22 in block DEMOG and record for 30 seconds
Item4=DEMOG.Q22[3], CONTINUE, 30
```

The Options section provides specifications that apply to the entire instrument. At this time, two options are supported, one to indicate whether recording is to be done for the entire instrument or only parts of it, and the other to indicate whether wave files should be given a unique time-stamped name so that the same item may be recorded multiple times in a case if needed for loops or other repetitive situations.

The Items section consists of 2 or more lines. There will always be a single ItemCount line, and there will be one or more Item lines after it. The Items section lists the questions which should be recorded, and for each item an “action” is specified. The available actions are

START – begin recording when this question is presented

STOP – end recording after this question’s response is entered

CONTINUE. – if recording is already on, continue to record

Any question which is not listed in the Items section will not be recorded, unless it is administered within a timed period.

Sometimes it is desirable to record for a specified length of time, rather than indicating on which question to start and stop. For example, if the path branches after a CARI question, a timer can be set to terminate the recording, instead of attempting to specify a stopping point on each branch. Item 1, Q5, in Figure 7 shows how a timer can be set, in this case for 20 seconds.

An advantage of using an external file to hold the list of CARI items is that it is easy to modify without changing the Blaise instrument itself. Programming the questionnaire can progress independently of decisions about which items to record. Tests and debugging of the recording process can be performed with recordings on questions early in the questionnaire and easily reached. This approach even permits modification of the CARI item list after an instrument has been released to the field without returning the laptop for updates and without affecting the data model. A new list can be transmitted to the laptop, replacing the original CARIRouter.ini, with additional items to record or different settings.

5. Constraints

Although the alien router technology can be very useful, as in the case of activating sound recording for CARI, there are times when it is not compatible with other features of Blaise. For example, the CARI alien router supersedes any other alien router within a Blaise block. For instance, the alien router supersedes usage of “lookup” and “classify” methods, because creating these in Blaise makes use of the alien router technology. If a “lookup” table is present in the same block with a user-defined alien router, or in a nested block, the lookup table will not work correctly. The dialog box which presents a value from an external database or classification type is not loaded by the DEP.

To avoid such conflicts, it may be necessary to redefine the block structure to segregate the fields using an external database or classification type the CARI alien router or other user-defined router.

6. Summary

By using a Blaise alien router with an external list of instrument items to record, CARI can be implemented quickly and easily, with little programming effort beyond the original development. A few lines of Blaise code are required to associate an alien router with a block when converting an existing survey instrument to one with CARI enabled. A simple text file serves as an external list of questionnaire items to record, allowing survey managers to amend the list during development or even after production data collection begins, without any effect on the data model of the instrument.

This technique provides support for easier quality assurance techniques which reduce burden on the respondents, field staff and surveying organization.

7. Acknowledgements

The authors would like to recognize R. Suresh, Albert Bethke and Phil Cooley of RTI International as the inventors of CARI in 1999. We would also like to acknowledge the contributions of Paul P. Biemer, RTI International, for developing and evaluating CARI methodology, the U.S. Census Bureau for research contributions, and many members of the Research Computing Division, Education Studies Division and Survey Research Division of RTI who have helped with suggestions, encouragement, design ideas, testing and conquering obstacles.

8. References

- Biemer, P.P., Hergert, D., Morton, J. and Willis, W.(2000), “The Feasibility of Monitoring Field Interview Performance Using Computer Audio Recorded Interviewing (CARI)”, *Proceedings of the American Statistical Association’s Section on Survey Research Methods*, pp. 1068-1073
- Statistics Netherlands, Statistical Informatics Department, P.O. Box 4000, 2270 JM Voorburg, The Netherlands.
- Suresh, R. (2005). “Web-Based Computer Audio Recorded Interview (Web-CARI).” Presented at the *International Field Directors and Technology Conference 2005*, Atlanta, GA
- Thissen, M. R., and Rodriguez, G. (2004), “Recording Interview Sound Bites Through Blaise Instruments”, *Proceedings of the International Blaise Users’ Conference*, pp. 411-423.
- Thissen, M.R, and Sattaluri, S. (2006), “Research and Development in Audio-Recorded Interviewing, Part II”, Presented at *The International Field Directors and Technologies Conference*, Montreal, Canada
- M. Rita Thissen, Sridevi Sattaluri, Emily McFarlane, and Paul P. Biemer (2007), “Evolution of Audio Recording in Field Surveys”, Presented at *The 2007 Conference of the American Association of Public Opinion Research*, Anaheim, CA