Blaise PlayBack and Recovery System

Youhong Liu & Gina-Qian Cheung (University of Michigan, USA)

1. Introduction

At the University of Michigan's Survey Research Center (SRC), the primary method for evaluating quality and providing feedback to field interviewers is for a supervisor or quality control specialist to review the completed questionnaires in Blaise while listening to an audio recording of the interview.

For this purpose on one study, SRC had developed a system to "playback" complete interview data. The technique utilized was to transmit each new keystroke read from a file to the Blaise Data Entry Program (DEP). This required that the Playback program (developed using Visual Basic 6) interact continuously with the DEP. The problem with this approach was that the two systems were sometimes out of sync, because the DEP executed rules while the VB program fed the data to the DEP. In addition, the system was further limited in that it was hard coded for a specific study.

The U.S. Bureau of the Census was interested using a generic version of such an audit trail, as were many Blaise users. They provided Michigan with funding and specifications for development such a system.

As a result, SRC developed a new generic Playback system. Everything in the new system is fully integrated, minimizing synchronization problems. For two of its functions (see section 3), it has an interface that replaces the DEP (with a "Mini Data Entry" routine) while retaining its essential elements, and provides control over both execution of rules and display of data. For a third function, it provides additional control for keeping the DEP synchronized with playback. This paper describes functions of the new system as well as techniques utilized in its development.

2. Audit Trail Information File (AIF)

First, the system assumes the use of the Blaise 4.7 audit trail DLL and an audit trail information file (.AIF), in order produce an ADT (audit trail) file in the format expected by the new Playback program. That is, the [MetaName].aif or audittrail.aif file must be present in the directory the system expects to find the audit trail DLL (Dynamic Link Library) during the interview. Either Blaise 4.6 or Blaise 4.7 may be used in data collection to achieve the desired results, but the Blaise 4.7 DLL and AIF are essential.

For playing back audit trails with the new SRC system, the setting for the options in the AIF should be as follows:

```
[AuditTrail]
LeaveField = 1
EnterField = 1
Action = 1
KeyStroke = 1
Classify = 1 or 0
Lookup = 1 or 0
ErrorDlg = 1
Forms = 1
```

When set to 1, the audit trail DLL will produce a separate audit file for each form

Extra = 1 or 0
FlushFile = 1
Mouse = 0 or 1
PreciseTiming = 1 or 0
KeyTranslation = 1

When set to 1, the audit trail will translate the virtual key code to a character or (when appropriate) to a description of the key

WriteLean = 0 or 1

This parameter can be specified as 0 or 1, WriteLean=1 option creates a more compact audit trail file. WriteLean=0 creates a more readable file. The program is written to process ADTs produced by both options so the user has flexibility according to his/her need.

KeyStrokeOneLine = 1
AuditFolder =
AuditTempFolder =

3. How Playback Works

There are three main components in the system. They are Playback, Recover, and Recover & RunDep. A description of each follows.

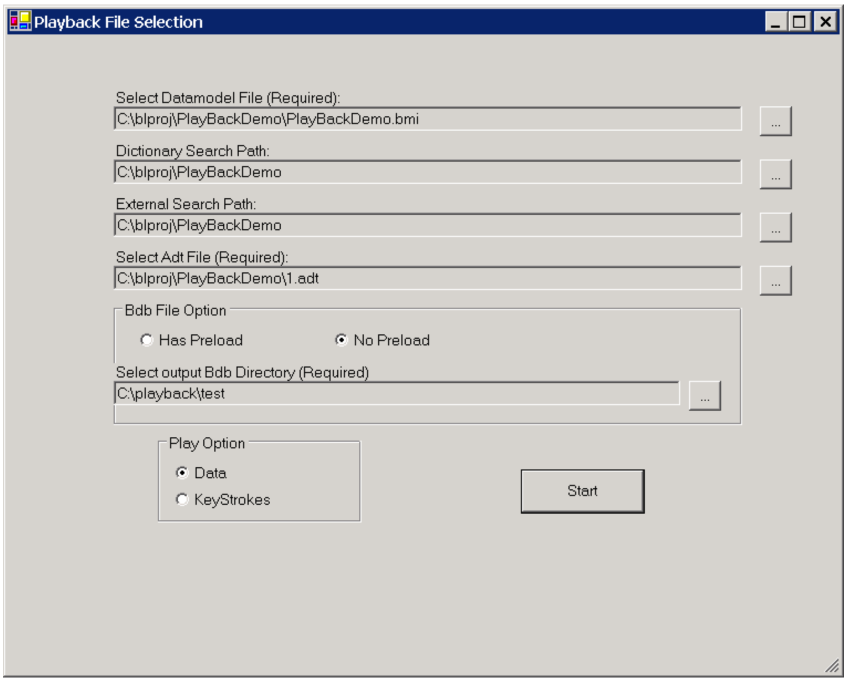
3.1. Playback

The first key screen in the playback option is the file selection screen (Figure 1). On this screen, the user needs to specify:

- The datamodel file (BMI) for the project;
- Dictionary Search Path; once you select a Datamodel, the Dictionary Search Path[1] will fill automatically with the path in which the datamodel was found; although, you may select another path;
- External Search Path; once you select a Datamodel file, the Dictionary Search Path[2] will fill automatically with the path in which the datamodel was found; although, you may select another path;
- Audit Trail File; an ADT file created using the previously selected datamodel or BMI file;
- Database or BDB file; (1) if preload data is expected, check “Has Preload” and select a BDB file location and file name; that BDB must contain the appropriate preload data, as it existed for the selected case and resulting ADT file the when the ADT file was created; (2) otherwise select “No Preload” and specify a directory where the program will create and store a new BDB file, named after the ADT, e.g. 123.bdb corresponding to123.adt; the preload database is important only if the Blaise instrument has fill, routing, and calculation logic that depends on preloaded variables.

[1] Indicates where dictionary files (typically those mentioned in the USES section of a Blaise data model) are located.
[2] Indicates the search path for external data files.

Figure 1: Playback Parameters Screen



Select Datamodel File (Required): C:\blproj\PlayBackDemo\PlayBackDemo.bmi

Dictionary Search Path: C:\blproj\PlayBackDemo

External Search Path: C:\blproj\PlayBackDemo

Select Adt File (Required): C:\blproj\PlayBackDemo\1.adt

Bdb File Option: ○ Has Preload ● No Preload

Select output Bdb Directory (Required) C:\playback\test

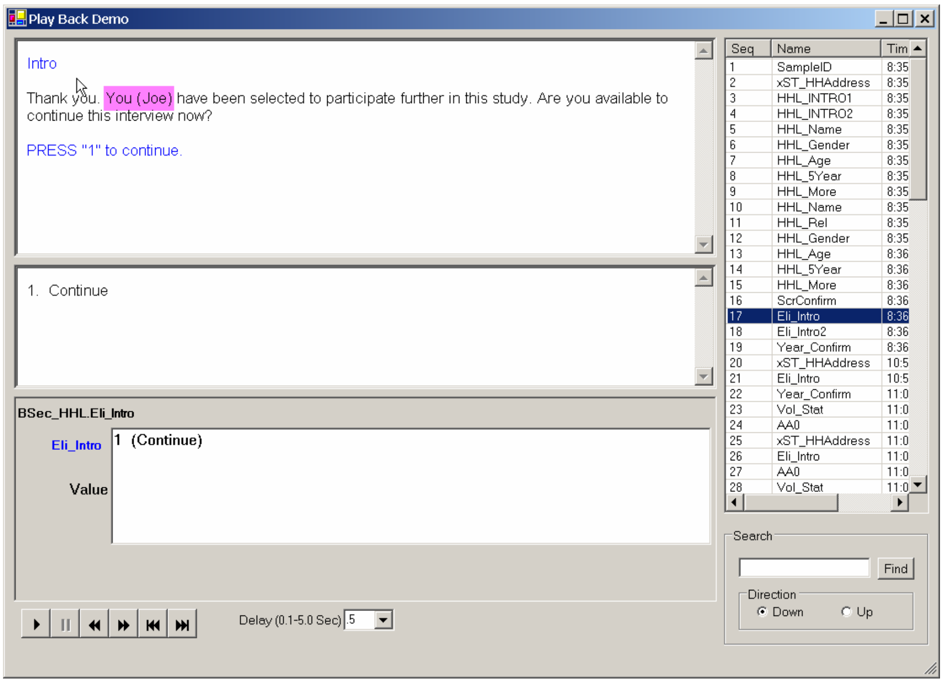
Play Option: ● Data ○ KeyStrokes

- Play Option; (1) select “Data” to play back the final values recorded in the ADT for questions; (2) or select “Keystrokes” play back all keystrokes and mouse actions, and to show the final values for questions.
- Starting Playback: The user presses the “Start” button (Figure 1) to begin playing back keystrokes for the selected ADT file. At this point, the program begins to process the adt file. It searches for an “Enter Field,” finds a corresponding “Leave Field” line, and saves all the information between these two lines in a structured array. Information stored and required for playback includes:

  FieldName – Blaise complete name of the field (conform the hierarchy in the data dictionary)
  ShortName – Blaise name of the field as defined in the surrounding block
  EnterLineNo – The “Enter Field” line number in the adt file for processing purpose
  LeaveLineNo – The “Leave Field” line number in the adt file for processing purpose
  IsLeaveFieldLine – Is there a “Leave Field” line in the adt block? Sometimes the “Enter Field” line does not have a corresponding “Leave Field”; in this case, use “Enter Field” value
  EnterValue – The value of the field when entering the field
  LeaveValue – The value of the field when leaving the field
  IsEnterForm – Is it the first field in a new form?
  IsLeaveForm – Is it the last field in the form?
  HasErrorDlg – Has soft check or hard check occurred in the field?
  ErrorStr – Error string saved for later processing
  Remarkchanged – Is Remark used in the field?
  ReMarkStr –Remark string saved for later process
  FormNO – Form Number
  Status – Is DK or RF entered for this field?
  Keys_() – Array of Key Strokes
  KeyCount – Number of “(KEY:)s” in the field block

  All of this information is stored for use during the playback. After the information is extracted, the program moves to the next screen.

Figure 2: Playback Mini Data Entry Screen



Intro

Thank you. You (Joe) have been selected to participate further in this study. Are you available to continue this interview now?

PRESS "1" to continue.

1. Continue

BSec_HHL.Eli_Intro

Eli_Intro: 1 (Continue)

Value

Delay (0.1-5.0 Sec): 5

Direction: ● Down ○ Up

Seq	Name	Tim
1	SampleID	8:35
2	xST_HHAddress	8:35
3	HHL_INTRO1	8:35
4	HHL_INTRO2	8:35
5	HHL_Name	8:35
6	HHL_Gender	8:35
7	HHL_Age	8:35
8	HHL_5Year	8:35
9	HHL_More	8:35
10	HHL_Name	8:35
11	HHL_Rel	8:35
12	HHL_Gender	8:35
13	HHL_Age	8:36
14	HHL_5Year	8:36
15	HHL_More	8:36
16	ScrConfirm	8:36
17	Eli_Intro	8:36
18	Eli_Intro2	8:36
19	Year_Confirm	8:36
20	xST_HHAddress	10:5
21	Eli_Intro	10:5
22	Year_Confirm	11:0
23	Vol_Stat	11:0
24	AA0	11:0
25	xST_HHAddress	11:0
26	Eli_Intro	11:0
27	AA0	11:0
28	Vol_Stat	11:0

Figure 2 shows a playback screen which has been named a "Mini Data Entry" screen because it resembles Blaise's Data Entry Program, displaying what the interviewer sees in the Form Pane and the Answer List Pane in a Question Text Window and Category Window, respectively. At the back end, the data is fed to the Blaise database and the rules are executed. The reason for a back-end database is that it can obtain proper fills for question and enumeration category texts. For example in Figure 2, "You (Joe)" is an example of fill text.

Components of this screen are:

a. Title, which is extracted from the selected BMI's description

b. Question Text Window, displays the field question text; the control used to display this text is a web browser control; the text is output to a temporary html file, and then is displayed in the web browser; the web browser control is chosen because it is easy to show different fonts and colors in the control, allowing the simulation of what the interviewer sees;

c. Category Window, displays the response options or categories, if the field is an enumeration type;

d. Keystrokes Window, displays all keystrokes and mouse actions entered at the field;

e. Value, displays the "Leave Field" value; if there is no "Leave Field" line associated with a field Playback just display the "Enter Field";

f. Navigation Buttons, which appear at the bottom of the "Mini Data Entry" screen, and control navigation through the instrument; they are:

▶ (Play) which starts systematic playback of data or keystrokes, moving through fields automatically in sequence,

⏸ Pause or Stop button; may be pressed at anytime to pause or stop execution,

⏩ Single step forwards; is used to move to the next field,

⏪ Single step backwards; is used to move to the previous field,

⏭ Jump to end; will move the last field in the list, and

⏮ Jump to beginning; will move to the first field in the list;

g. Field Name List, displays all field names in the instrument in the sequence in which they appear in the ADT; the available options are to:

- Double click a field in the list to jump to that field;

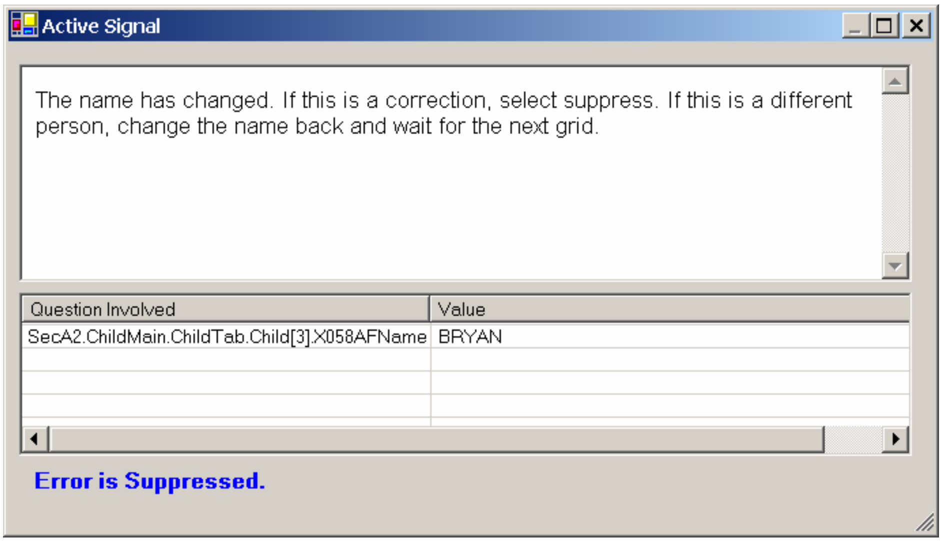
- Highlight a range of fields from the list; then press ▶(Play) to automatically go through fields selected, or press ▶▶to go through them one by one.

h. Delay Time, controls the delay in seconds between fields when ▶ (play) is pressed; the user can choose between a 0 and 5 second delay before moving to the next field;

i. Search, allows the user to jump to a field by searching for a field name. It supports partial and complete string searches;

If a soft (SIGNAL) or hard check (CHECK) is raised, or if the remark is accessed, Playback will display the contents of those windows (Figures 3 and 4). For a check, it will note the interviewer action (e.g., "Error is Suppressed").

Figure 3: Example of Soft Check Window and Its Contents for Displayed Field, with Resulting Interviewer Action

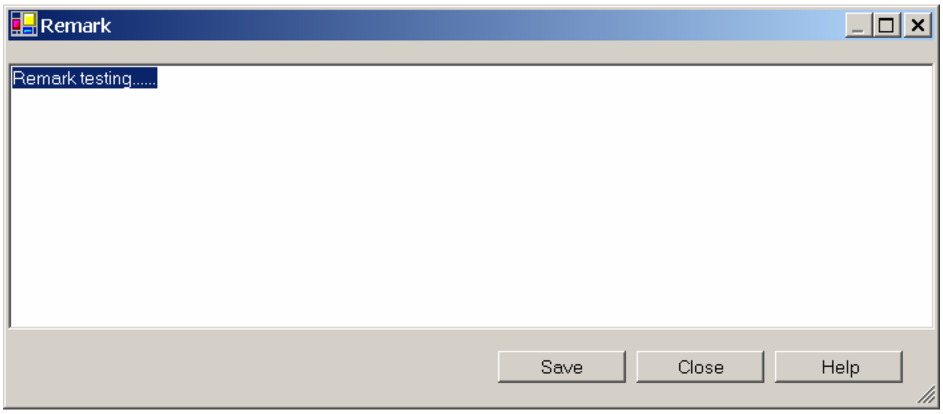


The name has changed. If this is a correction, select suppress. If this is a different person, change the name back and wait for the next grid.

Question Involved	Value
SecA2.ChildMain.ChildTab.Child[3].X058AFName	BRYAN

Error is Suppressed.

Figure 4: Example of Remarks Window and Its Contents for Displayed Field

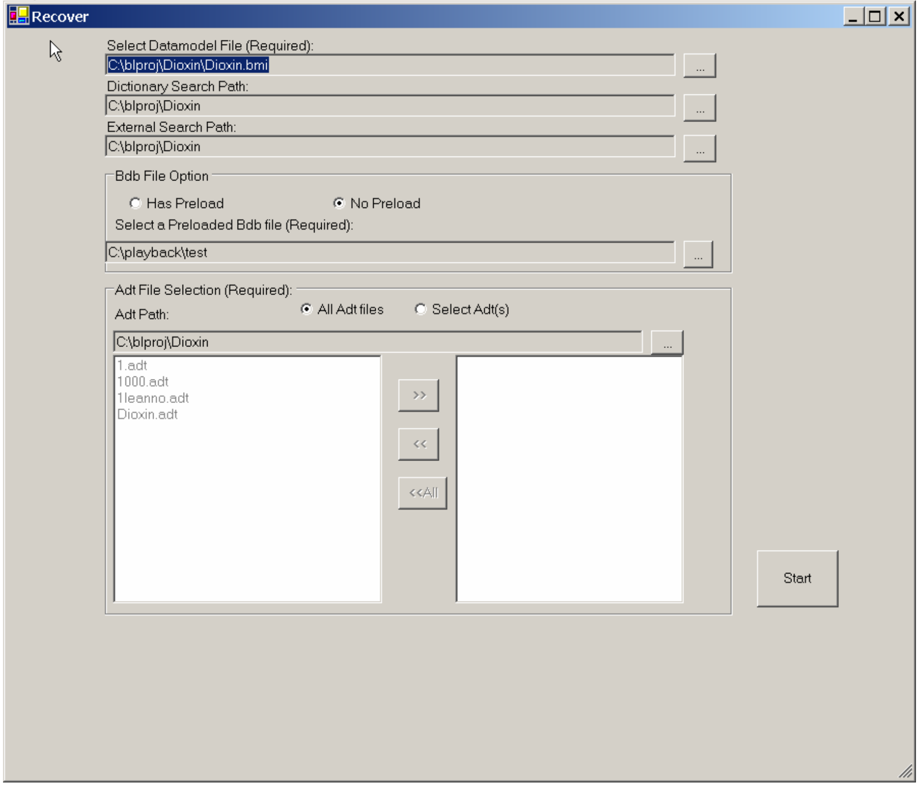


Remark testing......

3.2. Recover

The Recover component of the Playback system was designed to recover data from audit trail files. This generally would be desired if Blaise data files are lost or corrupted. Figure 5 shows the parameters screen for the Recover component. It is very similar to the Playback parameters screen. The main difference is that the user may choose more than one ADT from which to recover data. This is a much simpler process than that of Playback. The program will go through the selected ADT files and find the final value associated with a "Leave Field" line and save that value to a BDB file.

Figure 5: Recover Parameters Screen



Recover

Select Datamodel File (Required): C:\blproj\Dioxin\Dioxin.bmi

Dictionary Search Path: C:\blproj\Dioxin

External Search Path: C:\blproj\Dioxin

Bdb File Option

○ Has Preload ● No Preload

Select a Preloaded Bdb file (Required): C:\playback\test

Adt File Selection (Required):

Adt Path: ● All Adt files ○ Select Adt(s)

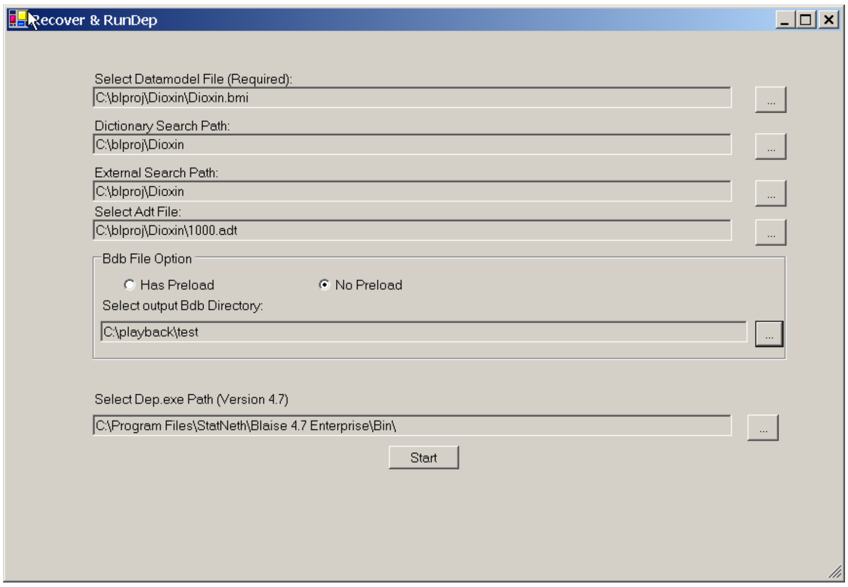
C:\blproj\Dioxin

1.adt
1000.adt
1leanno.adt
Dioxin.adt

3.3. Recover & RunDep

The third component of the Playback system is "Recover and RunDep." It differs from the other two in that it actually runs the Blaise DEP during the recovery process, and allows the user to "walk through the instrument by pressing [Enter] at each field as values are filled in. Figure 6 shows the parameters screen, with similar selections to be made. As with Playback, the user selects one ADT to recover and walk through. The program will first read the data from the ADT file and save final values to the database. The program will then launch the same case in Blaise Data Entry Program. This option is best for recovery of a suspended case from an ADT file and subsequent completion of the file. This may be desirable during the testing phase of development.

Figure 6: Recover and RunDep Parameters Screen



Select Datamodel File (Required):
C:\blproj\Dioxin\Dioxin.bmi

Dictionary Search Path:
C:\blproj\Dioxin

External Search Path:
C:\blproj\Dioxin

Select Adt File:
C:\blproj\Dioxin\1000.adt

Bdb File Option
○ Has Preload ● No Preload
Select output Bdb Directory:
C:\playback\test

Select Dep.exe Path (Version 4.7)
C:\Program Files\StatNeth\Blaise 4.7 Enterprise\Bin\

This component functions differently from the Recover component, in which the user can only save data fields. With Recover and RunDep, the program can also save the auxfield data and put those data into a parameter file. These data can then be passed from the parameter file to the DEP. Other parameters could also be written to this file. For example:

```
[DepCmd]
DataModel=T:\groups\TSG\tsprog\sysprog\SHARE\YLiu\NewPlayBack\Joel\Census\main\
        inst.bmi
MetaSearchPath=T:\groups\TSG\tsprog\sysprog\SHARE\YLiu\NewPlayBack\Joel\Census\
        main
ExternalSearchPath=T:\groups\TSG\tsprog\sysprog\SHARE\YLiu\NewPlayBack\Joel\Cens
        us\main
DataFile=T:\groups\TSG\tsprog\sysprog\SHARE\YLiu\NewPlayBack\Joel\Census\orig_bd
        bs\00000037.bdb
Key=00000037
GotoEnd=1
Assign==Front.Start_CP=2;Front.UnitStat_CP=1;Front.Dial_CP=1;Front.Intro_CP=1;Fron
        t.Ready_CP=1;
StartParallel=
```

In this example "GotoEnd=1" instructs the DEP to go to the first unanswered question on the route, and the "Assign" passes in the values of auxfields.

4. Error Handling

It is possible that the selected audit trail files would have errors that would cause the Playback program to stop running. To keep such situations to a minimum, the program checks for the following:

- The "Enter Field" field name must correspond with the field name of next "Leave Field;" this must be true for Playback only; this is hard check, and the program alerts the user that the ADT cannot be processed;
- The "Metafile name" must be the same as the selected datamodel name; this is a hard check, and the program alerts the user that the ADT cannot be processed;
- The "Metafile timestamp" should have the same date as the datamodel file's date. This is a soft check, which gives a warning message; the program will not stop running.

5. Randomization In the Blaise Datamodel

If the Blaise datamodel has a randomization field, and some logic depends on the value of this variable, problems may occur during playback and recovery. This is because Blaise may generate a different value during Playback than it generated during data collection. In this situation, data for these off-route fields may be lost. To avoid this, it is advised to preload randomized values.

6. Implications of Using an Alien Router Call

Alien Routers are external programs that Blaise calls to produce functionality and interface design that can not be achieved easily through Blaise directly. Because they are external programs, the audit trail DLL does not write any thing in the ADT file for the actions used in Alien Router. During playback/recover, keystrokes and mouse actions entered via an alien router call cannot be played or recovered. In order recover those values, the alien router needs to provide its own backup method. The use of such stored data needs to be managed (e.g., as preload) independently of the system described here.

7. Acknowledgements

The authors would like to acknowledge the contributions of Karen Bagwell, Rob Wallace, and Agatha Heesook, of the U.S. Bureau of the Census, and Eduardo Galvan, Sue Ellen Hansen, Joel Harvey, Jason Langfahl, Jason Ostergren, and Jim Hagerman of the University of Michigan. Their input on specifications, testing, and documentation have been invaluable.

8. References

Blaise OnLine Assistant (release 4.7.1000). (2005). Retrieved February 24, 2006 from World Wide Web:http://www.blaise.com/onlinehelp/

Langfahl, J. (2004). Replaying Blaise Audit Trail Files for Data Verification. Paper presented at the 9th International Blaise Users Conference, Gatineau, Québec, September.

Ostergren, J., Ash, R. (2004). Replaying ADK Files for Testing Applications. Paper presented at the 9th International Blaise Users Conference, Gatineau, Québec, September.