Reporting on Item Times and Keystrokes from Blaise Audit Trails

Sue Ellen Hansen, University of Michigan
Theresa Marvin, University of Michigan

1. 0 Introduction

Research has shown that trace and keystroke files are useful both for pretesting survey instruments and for evaluating interviewer performance (Couper, Hansen, and Sadosky, 1997; Couper and Schlegel, 1998). In Blaise, these files are called audit trails. In CAI systems that provide trace or audit trail files, they generally provide (1) the times that the interviewer entered and exited input fields or items, and (2) actions invoked at specific items, such as backing up, requesting online help, and recording or changing comments or remarks. Some systems, as does Blaise, also provide the option of recording every keystroke the interviewer presses during an interviewing session. Audit trail files with keystrokes are particularly useful in troubleshooting problems interviewers experience using computer assisted instruments and software.

In an analysis of trace files[1] from the 1997 National Health Interview Survey (NHIS) computer assisted personal interview (CAPI) instrument, Couper and Schlegel (1998) were able to identify a number of problematic survey items or screens, that is, screens with high levels of online help access or changed answers following back ups. These findings led to improved design of some of the NHIS screens. In an earlier study, Couper *et al.* (1997) examined keystroke files[2] from the 1993 CAPI Health Dynamics of the Oldest Old (AHEAD) CAPI study. They demonstrated the usefulness of keystroke files for exploring the effectiveness of interviewer training and for evaluating interviewer performance in the use of computer assisted interviewing (CAI) software and survey instruments (e.g., the extent to which interviewers access online help, successfully back up and change answers, press keys in error or make mistakes in the use of function keys).

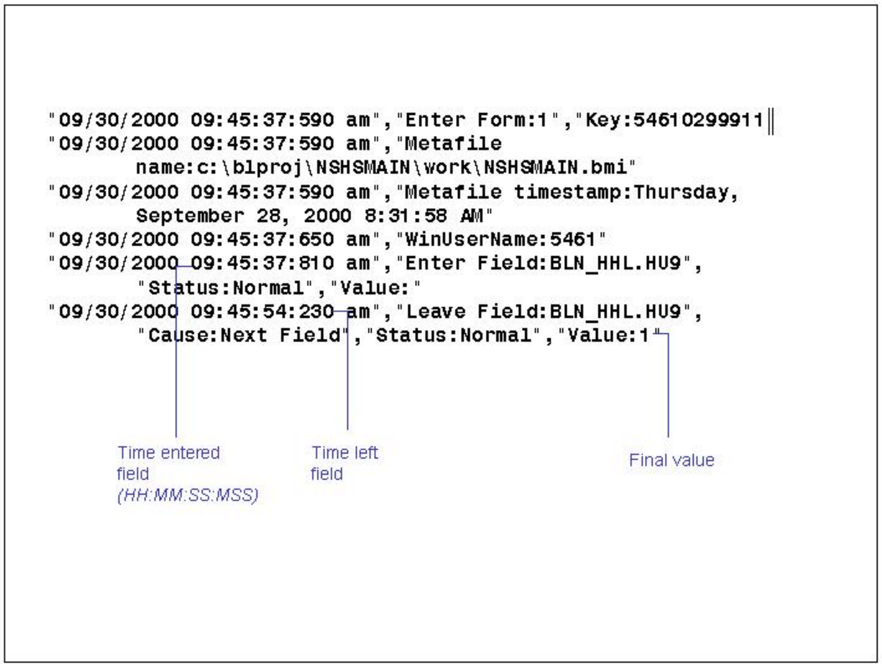
This paper discusses the Institute for Social Research's (ISR's) current use of Blaise audit trails. It describes how to set up Blaise to capture audit trails, and then provides an overview of the features of a program ISR uses to read the Blaise audit trails and report on item, block, and interview times. Examples of actual audit trail files and data provided by the reporting program are shown in Section 2, demonstrating how one can use audit trails to evaluate survey instruments and interviewer performance. The final section of the paper describes planned enhancements to the audit trail reporting program, and presents an example of additional information provided by a current beta version that reports on an expanded set of actions.

[1] The 1997 CAPI NHIS instrument was programmed in CASES. CASES trace files capture item level times, some commands such as backups and jumps, as well as the final stored response or value for each item accessed. Trace files also can be used to play back or review partial or complete interviews.
[2] The 1993 AHEAD CAPI instrument was programmed in SurveyCraft. SurveyCraft keystroke files capture item times, every interviewer keystroke, and can be used to play back interviews.

1.1 Blaise audit trails

Statistics Netherlands provides two Dynamic Linked Library (DLL) files that permit users to record audit trails for all Blaise cases, one that gives basic time data, and the other that records time data and keystrokes. ISR has edited both of these DLLs so that they provide time data to the millisecond (the Blaise default files provide this information to the hundredth of a second). The default DLL that captures keystrokes records numeric codes for all keystrokes, for which Statistics Netherlands has provided a translation table. ISR has used this table to modify the keystroke DLL so that it records short names for each keystroke (e.g., ENTR, BACK, etc.), making the audit trails more readable. It also has added the time when the first keystroke at an item is pressed[3]. The modified DLL files, *AuditkeyISR.dll* (time data only) and *KeyStrokesISR.dll* (time data with keystrokes) produce audit trails with the extensions *.ADT* and *.ADK*, respectively. As part of its sample management system, ISR captures interview lengths from Blaise audit trails[4], and thus requires all Blaise projects to create an audit trail for every Blaise case accessed, either a basic audit trail or an audit trail with keystrokes. Figures 1 and 2 show sample segments from each of these types of files.

Figure 1. Example from Audit Trail with Time Data Only

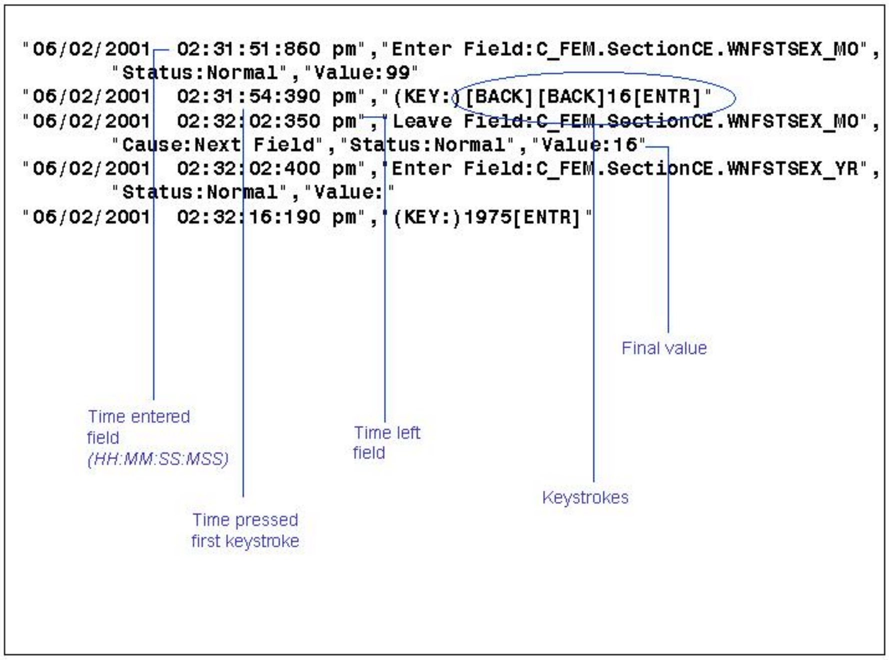


```
"09/30/2000 09:45:37:590 am","Enter Form:1","Key:54610299911║
"09/30/2000 09:45:37:590 am","Metafile
       name:c:\blproj\NSHSMAIN\work\NSHSMAIN.bmi"
"09/30/2000 09:45:37:590 am","Metafile timestamp:Thursday,
       September 28, 2000 8:31:58 AM"
"09/30/2000 09:45:37:650 am","WinUserName:5461"
"09/30/2000 09:45:37:810 am","Enter Field:BLN_HHL.HU9",
       "Status:Normal","Value:"
"09/30/2000 09:45:54:230 am","Leave Field:BLN_HHL.HU9",
       "Cause:Next Field","Status:Normal","Value:1"
```

[3] This allows for the calculation of the time between entering the field and the first keystroke, which could serve as a measure of response latency (Bassili, 1996).

[4] Although it is expected that survey instruments are programmed with essential time stamps, such as start of interview and end of interview, and perhaps key section times, ISR has found that audit trails are a more reliable source of survey timing data (Marvin *et al*., 2000).

Figure 2. Example from Audit Trail with Time Data and Keystrokes



```
"06/02/2001  02:31:51:860 pm","Enter Field:C_FEM.SectionCE.WNFSTSEX_MO",
       "Status:Normal","Value:99"
"06/02/2001  02:31:54:390 pm","(KEY:)[BACK][BACK]16[ENTR]"
"06/02/2001  02:32:02:350 pm","Leave Field:C_FEM.SectionCE.WNFSTSEX_MO",
       "Cause:Next Field","Status:Normal","Value:16"
"06/02/2001  02:32:02:400 pm","Enter Field:C_FEM.SectionCE.WNFSTSEX_YR",
       "Status:Normal","Value:"
"06/02/2001  02:32:16:190 pm","(KEY:)1975[ENTR]"
```

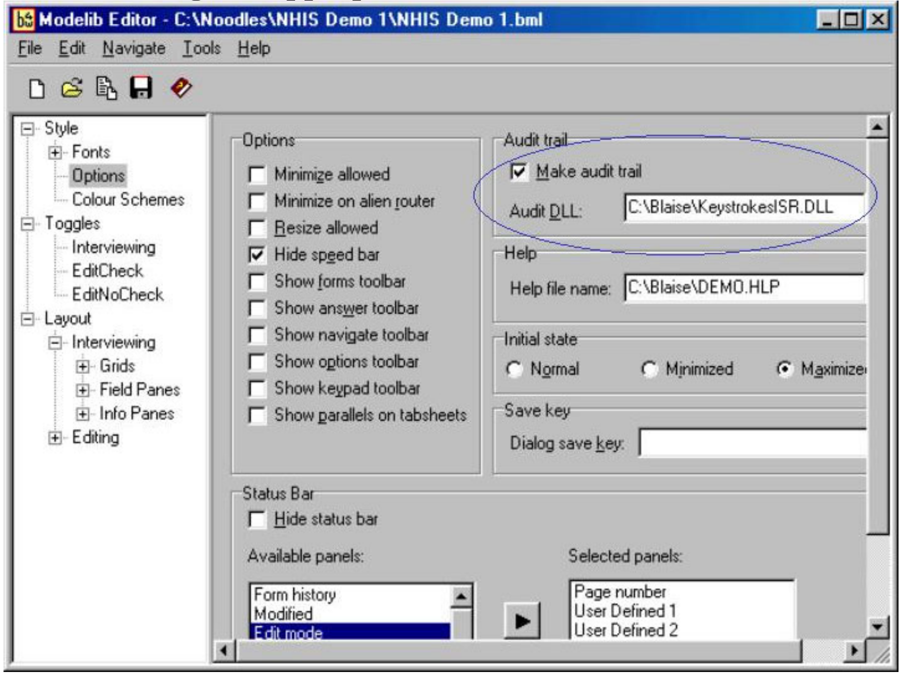
1.2 Setting up a Blaise project to create audit trail files

There are two things that ISR does to ensure that audit trails are captured for each case for a project. First, programmers modify default mode library settings to point to the correct DLL for the project. To confirm that the project collects audit trails with appropriate information, they check the mode library *Style Options*. They make sure that the *Audit Trail* box in the Options window is checked and that the file name for the audit trail format is *AuditkeyISR.dll* or *KeyStrokesISR.dll*. They also make sure that the appropriate directory *path* is specified, and that the selected DLL is in that directory (see Figure 3).

Second, programmers specify in the Blaise data model the variable that will be used to identify each Blaise interview or partial interview, which also uniquely identifies each audit trail file saved[5]. This is the "primary key" for the case, which generally immediately follows the SETTINGs in the data model. Figure 4 provides sample Blaise code that establishes the primary key used to identify each audit trail file.

[5] ISR sample management systems expect separate .BDB and .ADT (or .ADK, if audit trail has keystrokes) files for each case accessed.

Figure 3. Selecting the Appropriate Audit Trail DLL in the Mode Library



Options

- ☐ Minimize allowed
- ☐ Minimize on alien router
- ☐ Resize allowed
- ☑ Hide speed bar
- ☐ Show forms toolbar
- ☐ Show answer toolbar
- ☐ Show navigate toolbar
- ☐ Show options toolbar
- ☐ Show keypad toolbar
- ☐ Show parallels on tabsheets

Audit trail

☑ Make audit trail

Audit DLL: C:\Blaise\KeystrokesISR.DLL

Help

Help file name: C:\Blaise\DEMO.HLP

Initial state

○ Normal ○ Minimized ● Maximize

Save key

Dialog save key:

Status Bar

☐ Hide status bar

Available panels: Form history, Modified, Edit mode

Selected panels: Page number, User Defined 1, User Defined 2

Figure 4. Establishing a Primary Key in the Blaise Datamodel

```
DATAMODEL PrimKey "Primary Key Sample"

SETTINGS
   ATTRIBUTES = DONTKNOW, REFUSAL

PRIMARY
   SampleID
```

1.3 Processing audit trail files (ADT_Report.exe)

The program *ADT_Report.exe* was created in Delphi to transform the information in ADT and ADK files into a more useable format. *ADT_Report.exe* is designed either to run in batch mode whenever new Blaise data files are merged into a project master data file or to run interactively with a Graphical User Interface (GUI). When run in batch mode, the program creates a file that provides interview lengths (form total times).

Figure 5 shows the GUI for the interactive version of the program. It requires the following information:

- *Summary levels*, the levels at which times are reported:

Form	Interview
Block	Block, for instrument sections programmed as blocks (for example, block *Section A*)
Field	Question (for example, field *MarStat*)

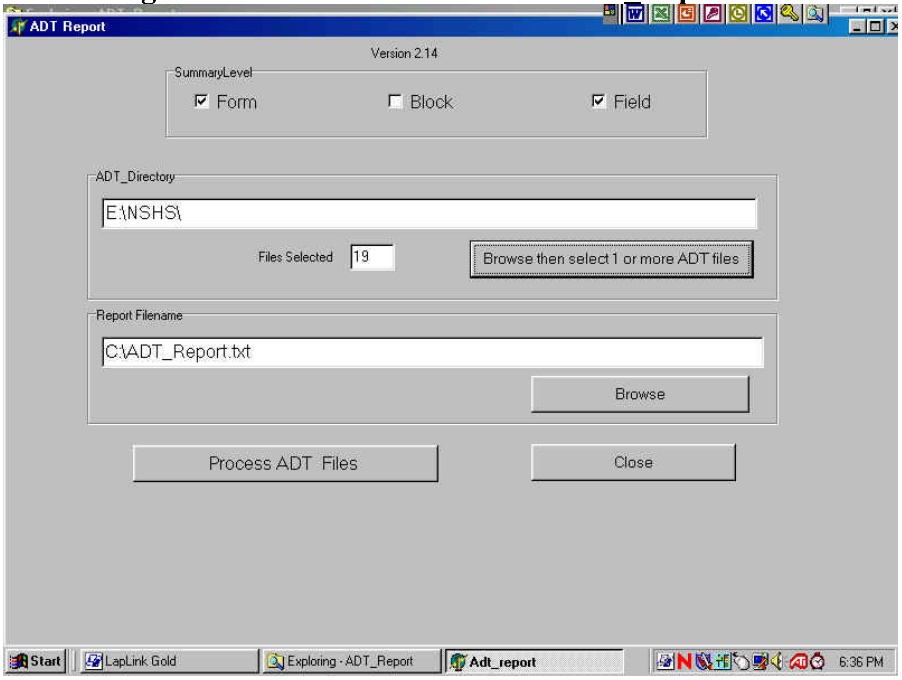
  By default, ADT_Report.exe reports times at the interview and question levels. Users who wish to report block times click on the Block box under Summary Level. To deselect Form, Block, or Field, they would click on the appropriate checked box.

- *The location and name(s) of one or more ADT or ADK files*. Users click on the *Browse then select 1 or more files* button, select the appropriate drive and folder, and select one or more files. By default the program looks in the folder from which it is running (*E:\NSHS\* in this example). After completing the selection, the program displays the selected folder and the total number of files selected (*19* in this example)

- *The location and name of the report file*. The *Report Filename* input box displays the location and name of the time report. By default this is *C:\ADT_Report.txt*. To change this, users click on the *Browse* button below this box, select the appropriate drive and path, and enter a new file name. In addition to changing the location of the file, users can also change the base name to reflect the project (e.g. C:\ADT_Report_ProjectName.txt). The filename entered here will be used as the base name for any future reports generated by *ADT_Report.exe* (see Section 3).

Finally, to generate the report, users click on the *Process ADT Files* button, and then click on the *Close* button to exit *ADT_Report.exe*.

Figure 5. Interactive Version of ADT_Report.exe



ADT Report

Version 2.14

SummaryLevel

☑ Form ☐ Block ☑ Field

ADT_Directory

E:\NSHS\

Files Selected 19

Report Filename

C:\ADT_Report.txt

1.3 Audit trail time reports

The report named in the *Report Filename* box reports at the level(s) specified in *Summary Level*. The program by default reports item and interview times. Interviewers or cases are identified as forms in Blaise. A report is an ASCII text file. Each line in the file is a record with at least the following fields, exclamation point delimited:

- CASEID Unique project-specific case ID (Blaise primary key or file name)
- FLD_SEQ Sequence in which the field appears in the interview
- FORM_VST Number of current visit to a form (for example, "2" if second visit across current and all prior sessions of the interview
- FLD_VST Number of current visit to a field (for example, "3" if third visit across current and all prior sessions)
- FLD_NAME Name of field (e.g., MarStat)
- FLD(H:M:S:Mss) Elapsed time between entering field and leaving field (hours-minutes-seconds-milliseconds)
- FLD(SSSS.mSS) Elapsed time between entering field and leaving field (seconds and milliseconds)
- BTW_FLD(H:M:S:Mss) Elapsed time between leaving field and entering next field (hours-minutes-seconds-milliseconds)
- BTW_FLD(SSSS.mSS) Elapsed time between leaving field and entering next field (seconds and milliseconds)
- ADJ_FLD(H:M:S:Mss) Sum of field time and following between time (hours-minutes-seconds-milliseconds)
- ADJ_FLD(SSSS.mSS) Sum of field time and following between time (seconds and milliseconds)
- EXIT "1" if last field in the form during a session; "0" otherwise

- METAFILE NAME — Complete name and path for the *project.bmi* file (version of the instrument)
- METAFILE TIMESTAMP — Metafile creation date and time
- WINUSERNAME — Windows user name (*login* name on network)
- REMARK_CLK — Number of times the interviewer opened the remark window
- REMARK_CHNG — Number of times the interviewer entered or changed a remark
- ERROR — Number of times there was a Blaise Action of Error Jump, Error Suppress or Error Escape.

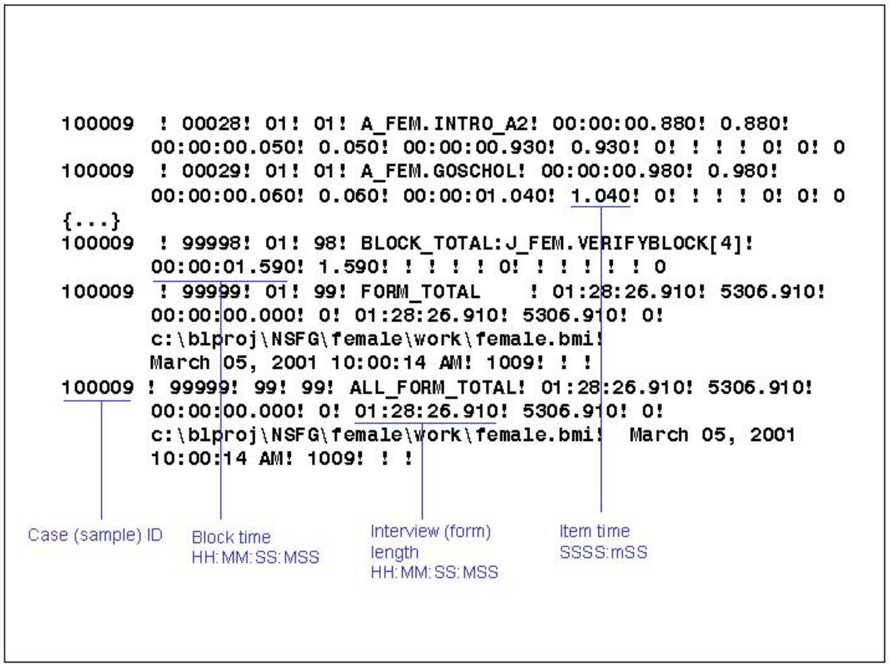
Metafile name, *metafile timestamp*, and *winusername* appear only on summary lines (for example, FORM_TOTAL). Flags indicating whether a function was invoked at an item (“1”; “0” if not) follow these fields. Current functions are from Blaise Actions that are output to the .ADT or .ADK file. Future versions of the program will output additional flags, for both Blaise actions and keystrokes pressed (see final section).

Item, block, and form times. Item times in the report represent time spent at an item for a specific visit to the item. Block times are the sum of times for all visits to items within the blocks. There are three times reported for every field, block and form: the field time, the time between that field and the next field (these times may or may not match), and an adjusted time (the sum of the prior two times). Each time is reported first in an hours-minutes-seconds-milliseconds format (HH:MM:SS:mSS), and then again in a seconds-milliseconds format (SSSS.mSS). The two formats are provided to accommodate the import requirements of various data management and analysis software packages. Generally ISR uses the adjusted times in analysis of audit trail data.

Since an interview may be conducted across multiple Blaise sessions, *ADT_Report.exe* reports both FORM_TOTALs (individual accesses of a form) and ALL_FORM TOTALs (sum of all accesses). A form total represents total time spent in an interview in one Blaise session, and the all-form total represents total time spent in an interview across all sessions. Thus, the program will report two form times and one all-form time for an interview that was suspended and later completed in a second session. The all-form times are at the end of the file. Figure 6 provides a sample report that shows item times, as well as block and form times (*ADT_Report.txt*).

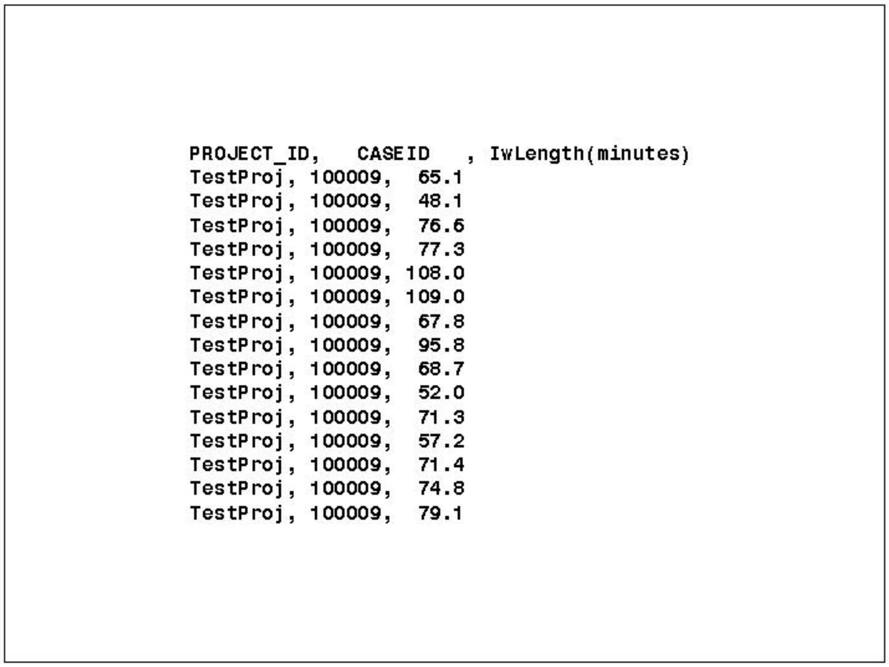
Interview length (form) report. The interview length is the sum of all entries to a form or case, across all visits to all items in the form. Figure 7 shows the interview lengths data file (*ADT_ReportIWLength.*txt), which is generated during batch processing. The interview lengths are then transferred to a table in ISR’s SurveyTrak sample management system. The case ID is based on the Blaise primary key (sample ID for ISR studies).

Figure 6. Sample from Data File with Item, Block, and Form Times



```
100009  ! 00028! 01! 01! A_FEM.INTRO_A2! 00:00:00.880! 0.880!
        00:00:00.050! 0.050! 00:00:00.930! 0.930! 0! ! ! ! 0! 0! 0
100009  ! 00029! 01! 01! A_FEM.GOSCHOL! 00:00:00.980! 0.980!
        00:00:00.060! 0.060! 00:00:01.040! 1.040! 0! ! ! ! 0! 0! 0
{...}
100009  ! 99998! 01! 98! BLOCK_TOTAL:J_FEM.VERIFYBLOCK[4]!
        00:00:01.590! 1.590! ! ! ! ! 0! ! ! ! ! ! 0
100009  ! 99999! 01! 99! FORM_TOTAL    ! 01:28:26.910! 5306.910!
        00:00:00.000! 0! 01:28:26.910! 5306.910! 0!
        c:\blproj\NSFG\female\work\female.bmi!
        March 05, 2001 10:00:14 AM! 1009! ! !
100009 ! 99999! 99! 99! ALL_FORM_TOTAL! 01:28:26.910! 5306.910!
        00:00:00.000! 0! 01:28:26.910! 5306.910! 0!
        c:\blproj\NSFG\female\work\female.bmi!  March 05, 2001
        10:00:14 AM! 1009! ! !
```

Figure 7. Interview Lengths Data Produced during Batch Processing of AuditTrails



```
PROJECT_ID,   CASEID   , IwLength(minutes)
TestProj, 100009,  65.1
TestProj, 100009,  48.1
TestProj, 100009,  76.6
TestProj, 100009,  77.3
TestProj, 100009, 108.0
TestProj, 100009, 109.0
TestProj, 100009,  67.8
TestProj, 100009,  95.8
TestProj, 100009,  68.7
TestProj, 100009,  52.0
TestProj, 100009,  71.3
TestProj, 100009,  57.2
TestProj, 100009,  71.4
TestProj, 100009,  74.8
TestProj, 100009,  79.1
```

1.5 Analysis

The current version of *ADT_Report.txt* provides the following information:

- Item and form times, and block times if requested
- Counts for each item of exits from the interview, remarks accessed, remarks entered or changed, and errors

In addition to automatically calculating and storing interview lengths, ISR uses audit trail files for the following purposes:

- Examining pretest form and item times, in order to determine whether survey instruments need to be shortened
- Identifying items at which at which interviewers overall may be experiencing a higher incidence of problems
- Identifying specific interviewers who may be experiencing an unusually high number of problems

The *ADT_Report.exe* data (exclamation point delimited) can be read into MS Access, SAS, SPSS, or any other data management or analysis software that can read ASCII delimited text. The following section provides examples from SAS analysis of the ADT_Report.txt file for .ADK files (times and keystrokes) from a large pretest for a study.

2.0 Findings

We analyzed data on audit trails (using the *KeystrokesISR.dll*) from a large pretest (619 interviews, 5,835 survey items, 30 interviewers). The pretest was for the National Survey of Family Growth (NSFG), Cycle 6, conducted during March through July 2001. There were two survey instruments, one for male respondents (n=310) and one for female respondents (n=309). Items with higher than average counts of item-level exits, errors (consistency checks), and average times were examined. We discuss each of these in turn, providing examples of potential problems in CAI instruments they can help identify.

2.1 High counts of exits from interview

The criteria for identifying pretest items with high interview exit rates were (1) that there were 200 or more visits and (2) that the exit rate for the single item (across all visits *to the single item* across all sessions in all interviews) was at least two standard deviations from the mean item-level exit rate *across all items*. Mean item-level exit rates were .05 (standard deviation = 1.8) for females and .07 (standard deviation = 2.2) for males. Table 1 lists the items that met both criteria[6].

[6] Less conservative selection criteria might have revealed additional problems. Any screen (other than the final screen) at which there is a relatively high number exits should be investigated for possible design or interviewer performance problems.

Table 1. Pretest Items with High Number of Exits from Interview

Item	Number of exits	Number of visits	Exit Rate (%)
Final screen in interview (female)	309	316	97.8
Final screen in interview (male)	310	318	97.5
Verify pregnancy outcome (female)	72	844	8.5
Introduction (section A female)	24	520	4.6
Introduction (section A male)	18	420	4.2

These items appear to represent three types of exits (1) normal exits at the end of the interview (male and female final screens); (2) getting into the first screen of the interview and then exiting without conducting the interview (introductions); and (3) abnormal termination of the interview at a verification item in the female instrument. There are reasonable potential explanations for the first two, the first benign, the second less so.

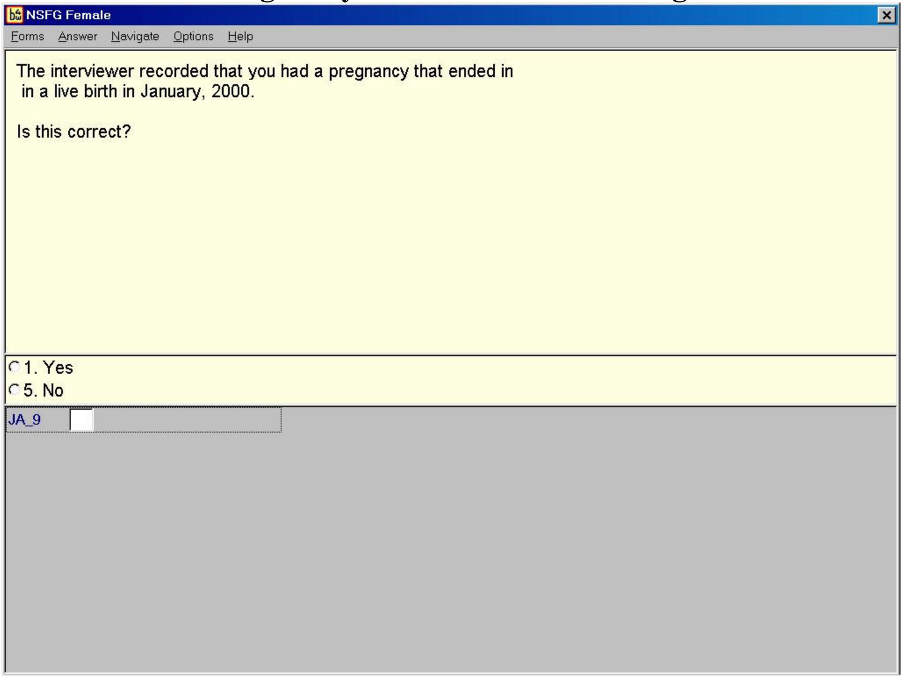
On the first, one might expect there to be a 100% correspondence between visits to the final screen in the interview, and exiting the interview. One explanation for less than 100% correspondence might be that interviewers in some cases backed up to review prior items before exiting. For example, the interviewer arrives at the final screen ("This concludes the interview. Thank you for your participation in this important study."), backs up for some reason to a prior item, revisits the final screen and then exits the interview. The first visit would not be flagged for exit, but the second would. This happens in less than 3% of the visits, and would not be considered a serious problem with the final screen *per se*. Examination of keystrokes in audit trails for cases involved would contradict or confirm the hypothesis about backing up to prior items. If supported, it would be useful to know what items were the targets of the back up sequence. Planned enhancements to audit trail reports would allow us easily to obtain such information (see Section 3).

On the second, it appears that interviewers arrived at the first screen in the interview, and more than 4% of the time exited without continuing. This suggests that interviewers sometimes get into the interview without actually knowing that they are able to proceed with the interview. This may be a training problem, or an aspect of the logistics of the decision to continue and move from the sample management system to the survey instrument that has not been anticipated. Additional investigation is necessary in order to identify the reasons for this unusually high number of exits at the introductory screens. These are also items at which there are unusually high item times. This may suggest that interviewers that exit do not make the decision to exit immediately after entering the screen (see discussion about item times below).

We explored why there would be very high numbers of visits to these items (520 for females and 420 for males). Since interviewers always revisit the introduction screen on reentry to cases previously exited, we calculated the total number of exits across all items in all cases, and found it to be 842. This accounts for approximately 90% of the 940 exits, which suggests that interviewers occasionally move forward through the first few items of the section and then return to the introduction screen. These early items ask the respondent to verify or provide additional information about household members already listed in a separate screener instrument. Interviewers may be returning to the introduction screen in an attempt to return to the screener, which was prohibited in this study. Detailed examination of the audit trail files would determine whether this is the explanation for the extra visits to the introduction screen. If this is the case, then the instrument designers could consider providing a mechanism for the interviewer to review a summary screen displaying the original roster, perhaps accessed via a function key.

The large number of exits at the verification item in the female instrument is more perplexing. These items occurred in a Blaise block used to verify the outcome of one or more pregnancies a respondent reported. The block occurred in a computer assisted self interview (CASI) section of the instrument that the respondent completed. The item (see Figure 8) included text fills, and by random assignment the respondent either read question text only or read question text while listening to text through earphones (Audio-CASI or A-CASI). Thus, for some respondents Blaise was also loading sound or *.WAV* files at this screen (a separate sound file for each segment of question text or text fill). Of the 72 verification screens from which interviews were abnormally terminated, 57% were text only and 43% were A-CASI.

Figure 8. Verification of Pregnancy Outcome Item with High Number of Exits



The interviewer recorded that you had a pregnancy that ended in
in a live birth in January, 2000.

Is this correct?

○ 1. Yes
○ 5. No

JA_9

Discussion with project staff revealed that respondents were required to type a “1” or “5” and press the [Enter] key at each pregnancy outcome verification item, and that they received reports that at least in some instances respondents experienced a delay after pressing [Enter]. Some respondents reacted by continuing to press the number and/or [Enter]s, and were eventually “thrown out of” the instrument. Thus, these 72 exits appear to represent abnormal terminations of the interview. Further investigation is necessary to determine whether these exits are related to the tailoring of text, the loading of multiple sound files, a software problem, how the verification block was programmed, or a combination of such factors related to a specific respondent. It is possible, through direct examination of the audit trails, to see exactly what keystrokes these respondents entered. This might assist in diagnosing the source of the problem at these screens.

In addition to providing information about survey items that may be problematic, audit trails can reveal information about interviewer behaviors. For example, we explored whether or not some interviewers were more likely than others to exit at the introduction item. Table 2 shows that only a few interviewers were responsible for the majority of exits at this item. Fourteen of the total of 30 interviewers exited at least once at the survey introduction screen. The percentage of the total exits each of the 14 interviewers was responsible for ranges from about 2.5% to 26%. Three interviewers alone (10% of the 30) were responsible for approximately 55% of the 42 exits.

Table 2. Individual Interviewer Percentages of Total Exits at Introductions

Interviewer Rank (high to low)	Number of cases	Number of exits	Interviewer exit rate (% of cases)	Percentage of total exits[a]	Cumulative Percentage
1	**18**	**11**	**61.1**	**26.2**	**26.2**
2	**29**	**7**	**24.1**	**16.7**	**42.9**
3	**9**	**5**	**55.6**	**11.9**	**54.8**
4	75	4	5.3	9.5	64.3
5	33	4	12.1	9.5	73.8
6	14	2	14.3	4.8	78.6
7	24	2	8.3	4.8	83.3
8	13	1	7.7	2.4	85.7
9	26	1	3.8	2.4	88.1
10	44	1	2.3	2.4	90.5
11	1	1	100.0	2.4	92.8
12	15	1	6.7	2.4	95.2
13	29	1	3.4	2.4	97.6
14	16	1	6.3	2.4	100.0
Total		42		100%	

[a] Individual percentages do not sum to 100% due to rounding.

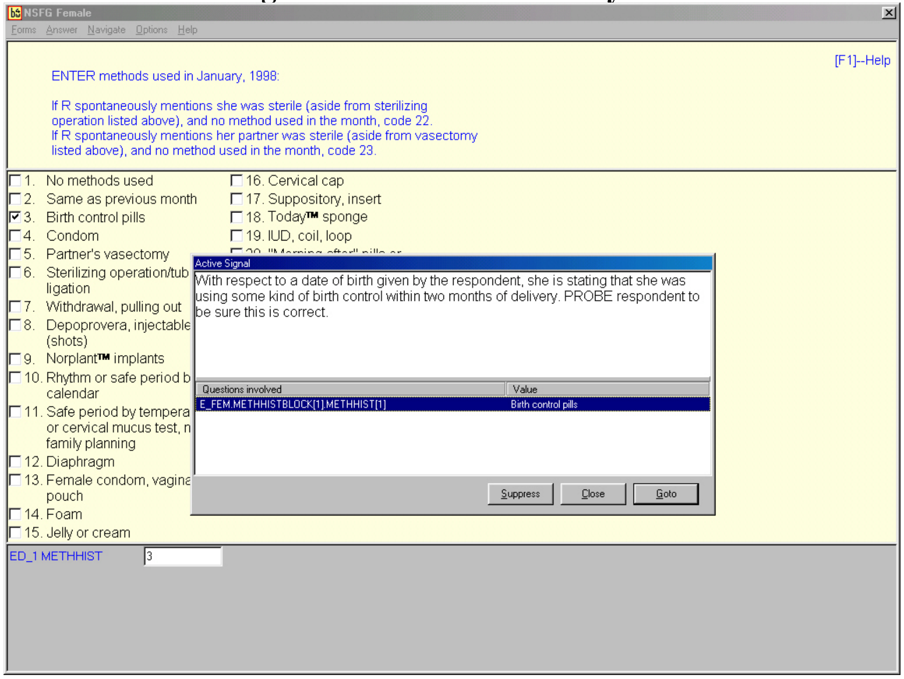
2.2 High counts of errors

ADT_Report.exe provides an "error" flag for each item visited, which is "1" if any Blaise error action occurred (Error Jump, Error Suspend, Error Escape), which are the result of programmed Blaise consistency checks. There were 4,525 consistency checks across all items in all cases. One item alone, a block of items asking for birth control history over a 36-month period had 1,332 consistency checks over 10,723 visits (see Figure 9). That is, 12.4% of the time one of the blocked items was asked, interviewers were asked to check the consistency of the response against other items in the series and elsewhere in the interview. This represents approximately 29% of the 4,525 consistency checks across all items in all interviews.

A great deal of care went into the crafting of custom error messages and programming a large number of consistency checks for this section, in an attempt to improve the quality of data. On such items there seems to be a tradeoff, however, between higher data quality and increased interviewer and respondent burden. The project has decided to redesign this section, placing the items in a calendar-like grid, in an effort to reduce both tedium and the number of consistency checks that occur, without reducing the number of programmed consistency checks.

Examination of mean case-level and interviewer error rates revealed that a few interviewers had much higher or lower mean counts of consistency checks per case. The mean count per case was approximately 8. One interviewer averaged 42 consistency checks per case over 5 cases, while another interviewer averaged zero consistency checks across eight interviews. Without further investigation it is difficult to provide an explanation for these outliers, which could be attributable to respondent characteristics, interviewer behavior, or both.

Figure 9. Birth Control Method History Item with High Number of Consistency Checks



ENTER methods used in January, 1998:

If R spontaneously mentions she was sterile (aside from sterilizing operation listed above), and no method used in the month, code 22.
If R spontaneously mentions her partner was sterile (aside from vasectomy listed above), and no method used in the month, code 23.

□ 1. No methods used
□ 2. Same as previous month
☑ 3. Birth control pills
□ 4. Condom
□ 5. Partner's vasectomy
□ 6. Sterilizing operation/tub ligation
□ 7. Withdrawal, pulling out
□ 8. Depoprovera, injectable (shots)
□ 9. Norplant™ implants
□ 10. Rhythm or safe period b calendar
□ 11. Safe period by tempera or cervical mucus test, n family planning
□ 12. Diaphragm
□ 13. Female condom, vagina pouch
□ 14. Foam
□ 15. Jelly or cream
□ 16. Cervical cap
□ 17. Suppository, insert
□ 18. Today™ sponge
□ 19. IUD, coil, loop

With respect to a date of birth given by the respondent, she is stating that she was using some kind of birth control within two months of delivery. PROBE respondent to be sure this is correct.

Questions involved	Value
E_FEM.METHHISTBLOCK[1].METHHIST[1]	Birth control pills

ED_1 METHHIST 3

2.3 High item-level times

Mean item time across all items was 12.1 seconds for female respondents, and 11.4 seconds for male respondents. However, using the same criteria as for exit rates (200 visits and two standard deviations above mean), we identified approximately 20 items with relatively high mean times, ranging from a high of 3 minutes 32 seconds to 38 seconds. Some of the items with high mean times parallel those found in the analysis of high exits: (1) the introduction screen (3 minutes 32 seconds mean time in the female instrument and 3 minutes in the male instrument); final screen (1 minute 30 seconds in female instrument and 1 minute 53 seconds in the male); and verify pregnancy outcome (approximately 1 minute in female).

Some interviewers were relatively higher or lower than the means on interview length. For example the mean female interview length was 2 hours and 13 minutes. The means for two interviewers (out of 30) were two and one standard deviation above the mean (4 hours and 52 minutes for 1 interview, and 3 hours 3 minutes across 3 interviews). The means for two other interviewers were at least one standard deviation below the mean (mean of 1 hour 33 minutes across 15 interviews, and 1 hour 24 minutes across 12 interviews, respectively). Further analysis could investigate the general level of correlation among certain behaviors (for example unusually high exits, item times, and consistency checks. Direct examination of audit trail files for outlier interviewers, cases, and items could help explain some of the high item times and identify questions to delete from the pretest instrument or questions to redesign in an effort to improve usability and interviewer efficiency. These findings also suggest that interview length may go down with experience on the survey and increases in number of interviews taken.

In summary, audit trails provide a wealth of automatically collected data that that can be used to identify potential problems at the question, interviewer, and case levels. With the added capability of capturing keystrokes as well as item times, Statistics Netherlands has now made it possible to learn a lot about some of the sources of problems, and the situations in which they occurr (including item-level events and behaviors, and the sequences in which they occur). Improved reporting programs will make it possible to quickly report on these data and identify key problems in survey questions and interviewer performance.

3.0 Future steps in the development of audit trail analysis tools

ISR programmers have begun modifications to the basic *ADT_Report.exe* in order to report on additional functions or interviewer actions that occur at each item, and we are currently testing a beta version. In this version, function counts *at the item level* are:

- SUSP	Number of times the interviewer suspended or exited the form normally
- ABSUSP	Number of times the interviewer suspended or exited the form abnormally
- TOTSUSP	Total number of times the interviewer suspended the form, either normally or abnormally
- C_X_SUSP	Number of times the interviewer form suspended using the CTRL_X function
- REMCLK	Number of times the interviewer opened the remark window (via menu or F2)
- F2_REMCLK	Number of times the interviewer opened the remark window by pressing F2
- REMCHNG	Number of times the interviewer entered or changed a remark
- QHELP	Number of times the interviewer accessed question-level help (via menu and F1)
- F1_QHELP	Number of times the interviewer accessed question-level help with F1
- BLAISEHELP	Number of times the interviewer the interviewer accessed the Blaise Help function
- NEXTLANG	Number of times the interviewer selected “next language” (via menu or F12)
- F12_NEXTLANG	Number of times the interviewer selected “next language” via F12
- PREVLANG	Number of times the interviewer selected “previous language”
- C_L_LANG	Number of times language option box was opened at this field

These counts reflect recorded Blaise actions, pressed keystrokes, or a combination of both. Thus, they are most useful when using the *KeystrokesISR.DLL* to generate audit trails. The ability to distinguish between types of function access (e.g., menu versus function key) allows us to evaluate the effectiveness of training and interviewer performance in use of specific functions. For example we examined the frequency of help access for these pretest interviews and found that help was accessed very little across all items in all pretest interviews (item-level mean for females was. 04 and for males .08). It would be useful to explore whether rates of help access were higher at the beginning of a study, whether some interviewers access help more than others, or whether some items have much higher rates of help access. As with the analysis discussed in Section 2, examination of function use in this way may reveal problems or weaknesses and lead to improved design of screens, on-line help, training, and so on.

In the future we will add additional counts to the item-level data and new reports. One report we expect to create will be a "back up" report that will include items that were the target of a backup sequence (one or more keystrokes were pressed to return to a prior item), the source item (from which the backup sequence was initiated), the number of items backed up through, and the original and final data values for the target item. ISR was able to generate such reports for CASES and SurveyCraft, which were used to identify problematic survey questions (see, for example, Couper and Schlegel, 1998). Another report we hope to create will report on item-level error signals and checks, and items gone back to, with original and final answers at those items. A third report we have considered creating is a report on the actions and times associated with the use of multimedia files, such as playing and replaying video and audio files. Finally, we plan to report on item-level response latencies, and case- and interviewer- level response latency times, based on the difference between the times fields are entered and the time the first keystroke is pressed.

Current versions of the audit trail reporting program and DLLs have been placed in the International Blaise Users Group (IBUG) *E-Room*. The internet address is http://eroom.isr.umich.edu/eroom/isrrooms/Blaise, and the login and password are *Blaise User* and *bclub*, respectively (case sensitive). Access does not require that you have licensed *E-Room* software. As new versions become available, we will place them in the *E-Room*. Questions may be sent to sehansen@umich.edu or tmarvin@isr.umich.edu.

4.0 References

Bassili, John N., and Stacey B. Scott (1996). Response Latency as a Signal to Question Problems in Survey Research. *Public Opinion Quarterly 60*(3):390-399.

Couper, Mick P., Sue Ellen Hansen, and Sally Sadosky (1997). Evaluating Interviewer Use of CAPI Technology. In Biemer, Paul, *et al., Survey Measurement and Process Quality*. New York: John Wiley & Sons.

Couper, Mick P., and Jay Schlegel (1998). Evaluating the NHIS CAPI Instrument Using Trace Files. *Proceedings of the Survey Research Methods Section*. Alexandria, VA: American Statistical Association.

Marvin, Theresa, *et al.* (2000). Collecting Timing Data in Blaise. Report submitted to the U.S. Bureau of Labor Statistics (Task Order #22, Contract Number J-9-J-0025), June.