Implementing Audio Computer Assisted Self Interviewing in Basil

Thomas Spaulding, U.S. Census Bureau

1 Introduction

The National Health Interview Survey (NHIS), which is sponsored by the National Center for Health Statistics (NCHS), is the main source of information on the health of the non-institutionalized population in the United States. The survey provides information about the prevalence and distribution of illness, its effects in terms of disabilities and chronic impairments, and the kind of health services people receive.

As early as 2004, the NCHS had talked to the U. S. Census Bureau about the possibility of using Audio Computer Assisted Self Interviewing (ACASI) technology to ask questions about sexual orientation in the NHIS instrument. The NCHS was aware of the sensitive nature of the questions and wanted to use ACASI to create a private environment for the respondent to answer the questions.

In ACASI, the interviewer does not have to read questions aloud to the respondent nor does ACASI require the respondent to provide a verbal answer to the interviewer. An ACASI environment provides the respondent with a device (i.e., some sort of computer) with which to complete a questionnaire. The device displays questions and answer lists on a screen while the respondent listens to audio recordings of the same questions and answer lists through headphones. The respondent answers each question by pressing the appropriate computer key (or touching the screen appropriately if the device has touch screen capability). This approach would create the type of private environment that the NCHS wanted for its respondents.

In September 2010, the Question Design Research Laboratory (QDRL) at the NCHS began discussions with the Census Bureau about conducting an ACASI pilot project. The QDRL 's initial plan was to have the Census Bureau's Field Representative (FR) who conducts the NHIS interview provide the respondent with a small touch screen device (e.g., iPad, iPod, or smart phone) on which the ACASI portion of the interview would be administered. The device would present just the question and response options on the screen, and respondents would only need to touch the screen to select the desired response. The QDRL had initially envisioned an ACASI questionnaire that was a multimedia application using sound, images, and video.

Unfortunately, the time frame for developing and deploying the pilot project precluded the use of such a device. There was not enough time to go through the procurement and security processes required to place a new device in the field. As a result, the pilot project would have to run on the existing FR laptops. Considering this constraint, the authors at the U. S. Census Bureau recommended to the NCHS that the pilot project use the ACASI functionality available in Blaise. After some investigation, the NHCS survey methodologists decided that the Blaise user interface was more complicated than what they wanted to use, and it could not be customized sufficiently to meet their needs. The NCHS methodologists wanted a user interface that would be simple in appearance and easy for respondents with limited technical knowledge to use when answering the questions in the ACASI module. Following discussions with Westat and Statistics Netherlands, the NCHS researchers decided to develop the ACASI module in Basil which is better suited for self interviewing situations.

This paper discusses some of the technical, methodological, and organizational challenges encountered while the NCHS and the U. S. Census Bureau worked on the ACASI module. It also discusses some of the advantages and disadvantages of using Basil for a project of this nature and some experiences from the first field test.

2 ACASI Project Requirements

The NHIS ACASI project was a new experience for the Census Bureau in a couple of ways. This was the Census Bureau's first attempt to field an ACASI application, and our first significant experience with Basil. In the past the Authoring Staff at the Census Bureau had researched ACASI technologies, but no inquiry from any of our customers had ever reached the point where we actually had to develop and field an ACASI application. This project was also the first time that the sponsoring agency wanted to develop a prototype of the application on its own and then turn the application over to the Census Bureau to complete. The NCHS survey methodologists and the NCHS programmers in the QDRL wanted to work together closely during the initial stages of the project so they could experiment with the ACASI technology and rapidly develop and test various methods of asking sensitive questions. Ultimately, the QDRL developed a "proof of concept" prototype of the ACASI module. The QDRL turned that prototype over to the Census Bureau's Authoring Staff, and it was the Census Bureau's task to complete the prototype and integrate it into the NHIS instrument so it would be ready for a field test.

2.1 The ACASI Prototype

While working on the prototype of the ACASI module, the QDRL's major concern focused on usability. The QDRL wanted to keep the user interface as simple as possible for the respondent. As a result, the QDRL decided to keep the number of keys the respondent had to use to a minimum. For most questions this meant the respondent only needed to know how to use three keys. The exceptions were questions that asked the respondent to key in a number or a string of text. There were only a couple of questions like this in the ACASI module, but they required the use of the letter and number keys on the laptop's keyboard.

The three main keys used in NCHS's ACASI prototype were the Enter key, the Space Bar, and the Caps Lock key. (The Caps Lock key was later replaced by the Tab key in the version of the module used in the field test.) All three keys were covered by color coded stickers to make them readily recognizable. The Enter key became the Green button, the Space Bar became the Red Circle key, and the Caps Lock key became the Orange button. (When the Caps Lock key was replaced by the Tab key, its color also changed from orange to blue.)

The Green button (i.e., the Enter key) saved an answer and moved to the next question, the Orange button (i.e., the Caps Lock key) backed up one screen in the instrument, and the Red Circle key (i.e., the Space Bar) allowed the respondent to scroll through the answer list and return to the question (if needed). Figure 1 is a screen shot from an early version of the NCHS's prototype. It is a depiction of one of the instructional screens used in that particular version of the ACASI module. This screen did not make it into the final version of the prototype, but it does show what the laptop keyboard looked like with the color coded keys.

Figure 2 is from the same prototype. (This particular question also failed to make it into NCHS's final prototype.) Figure 2 shows how the respondent can use the Red Circle key (i.e., the Space Bar) to scroll through the answer list to select a response. When the respondent first lands on the screen, the focus falls on the question text, and the ellipse is not displayed. Pressing the Red Circle key moves the focus to the first answer in the answer list which is then encircled by the red ellipse. Each time the Red Circle key is pressed, the focus moves to the next answer in the list until the last answer option is reached. Pressing the Red Circle key while on the last answer option moves the focus back to the question text.

Figure 1- Instructional Screen from the Prototype Showing Color Coded Keys on Keyboard

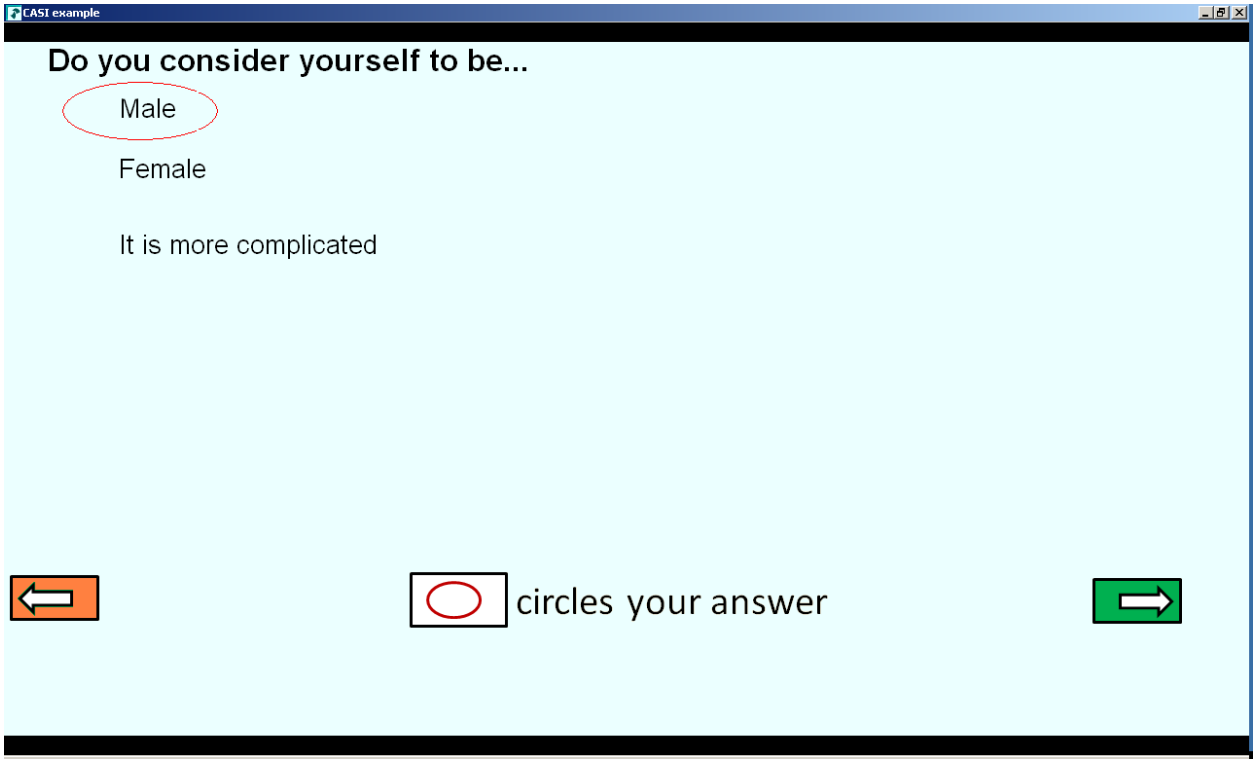

Figure 2 - Selecting a Response in the Prototype by Pressing the Red Circle Key

2.2 High Level ACASI Requirements

The NCHS provided the Census Bureau with its final ACASI prototype in September 2011, and this application served as the basis for the requirements that the Census Bureau would use to complete the ACASI module for the field test.

The subject matter specialists in the Census Bureau's Demographic Surveys Division (DSD) documented the following high level requirements.

The specialized, color coded keys to use in the field test instrument would be:

- Green Arrow key (i.e., Enter key) = Record an answer and go to the next question
- Red Circle key (i.e., Space Bar) = Scroll through answer categories and question text
- Blue Arrow key (i.e., Tab key) = Back up one screen

Other global instrument functionality included:

- To hear the audio for a question repeated, the respondent will scroll through the answer categories until the focus returns to the question.
- The instrument should allow a blank answer at each question. The respondent does not have to select an answer category in order to go forward to the next screen (i.e., allow Empty for each screen).
- For the first field test, once the respondent completes the ACASI section, neither the respondent nor the FR will be able to back into the ACASI section. There will be a screen at the end of the ACASI module that tells the respondents that they will not be able to go back and change their answers.

The NHIS Blaise instrument must launch the ACASI module.
Data collected in the ACASI module must be stored in the NHIS Blaise instrument's database.

- Audio recordings will be in .mp3 format. The audio files will be recorded using a human voice. (The NCHS prototype had used text-to-speech technology.)

2.3 Requirements for Field Testing the ACASI Module

The NCHS requested that the Census Bureau conduct three field tests of the ACASI module integrated with the NHIS instrument.

The Census Bureau conducted the first field test in November 2011. This was a small scale test that used only 6 FRs to administer the questionnaire. The goal of the test was to collect 50 completed cases. This test focused on the programming of the ACASI portion of the questionnaire and on the procedures that the field staff would use to administer the ACASI portion of the interview. Some production procedures were not employed to expedite the test. For example, the FRs only had to collect data for one adult, and it was not necessary for them to collect data for the entire household. The test also provided an opportunity to gauge initial reactions to the instrument from both the FRs and the respondents. Some experiences from this test are discussed later in this paper.

The second test that the NCHS requested requires a sample of 1,100 cases and will employ about 100 FRs. Its goal is to collect 500 completed cases. The Census Bureau will conduct this test in April 2012, and the test will utilize all the field procedures used in production for the NHIS. The NCHS hopes that this test will provided a realistic reflection of the field effort required to obtain completed ACASI interviews and of respondent acceptance of ACASI.

The third test is scheduled for July 2012. At this time, it is not known how many FRs will be participating in this test, but the goal is to collect 5,000 completed cases. There will be a split sample of 11,000 cases. About 60% of the sample will get the ACASI treatment. The other 40% will use flash cases to respond to the sensitive questions. The NCHS will use the results from this test to determine whether to implement the ACASI questionnaire for production data collection.

3 The ACASI Module for the 50 Case Test

The ACASI module for the 50 case test included instructional screens, sexual orientation questions, financial concern questions, questions on sleep, mental heath questions, and alcohol use questions. This section of the paper describes the basic organization of the Basil datamodel, the type of questions

administered in the module, and the basic approach the authors used to implemented a question in Basil.

3.1 Structure of the ACASI Module

The ACASI module starts with the required Basil application section. This section differentiates a Basil application from a Blaise instrument and is used to specify global information about the layout and behavior of the instrument. For example, the NHIS ACASI module's application section specifies that the application uses the entire screen (and that it is not divided into panels). The application section also specifies what actions should take place when the Basil questionnaire starts and when it closes, it identifies the location of the Maniplus procedures that are used by the Basil application, and can associate an a event with specific key.

An AUXFIELDS section follows the application section. It defines all the Basil question sections that the ACASI module uses. A Basil question section contains the information required for displaying one question. In the NHIS ACASI module, the question sections are used to represent both the questions and the answer choices. Each question and each response option in a question's answer list are separate AUXFIELDs, and each AUXFIELD's Field description contains its own unique Basil question section.

The AUXFIELDS section is then followed by the FIELDS section that defines the actual variables into which a respondent's answers will be stored. The data in these Fields will later be copied to the Blaise instrument's database. A RULES section provides the logic for how the Fields are to be processed in the ACASI module. (As will be noted later, the ACASI module makes considerable use of Maniplus procedures to perform event handling that assists in navigation, range checking, and data storage during the ACASI interview.) Finally, the LAYOUT section specifies which of the AUXFIELDs containing Basil question sections appear together on the same page.

3.2 Types of Questions in the ACASI Module

The ACASI module uses only three types of "questions." The first type of question is actually an instruction. These instructional items provide the user with information about using the ACASI questionnaire. Most of these appear at the beginning of the questionnaire. There are also questions with an answer list (many of which include 'Don't know' and 'Refusal' options), and there are a few questions that ask the respondent to enter either a numeric or text response into an input box. Examples of all three types of questions appear in the following three screen shots.

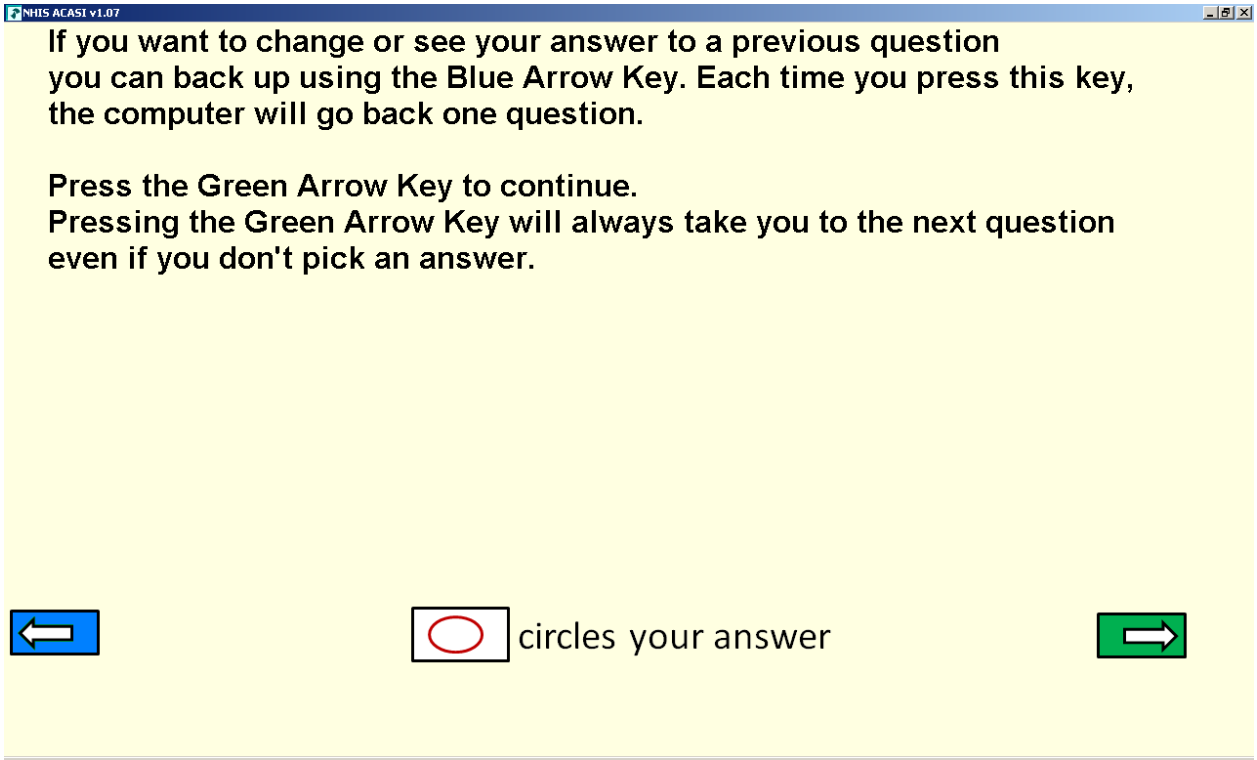

Figure 3 - An Instructional Screen

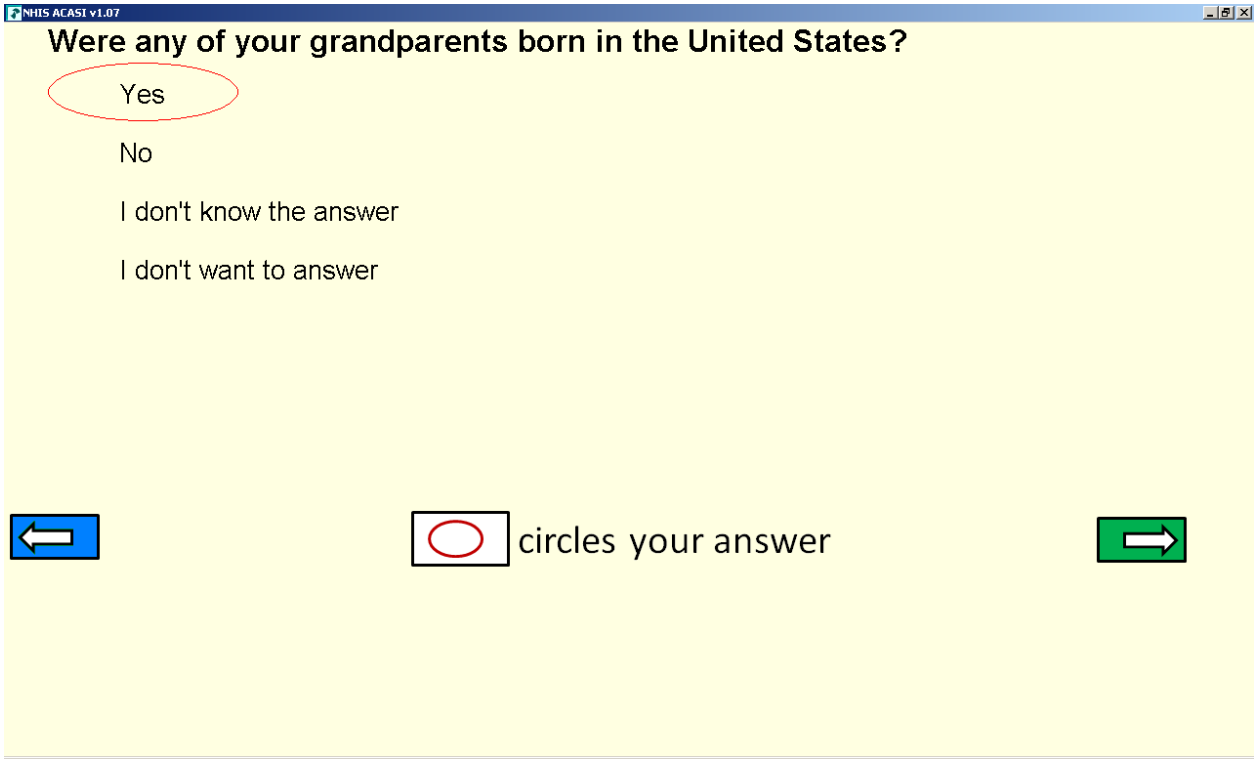

Figure 4 - A Question with an Answer List

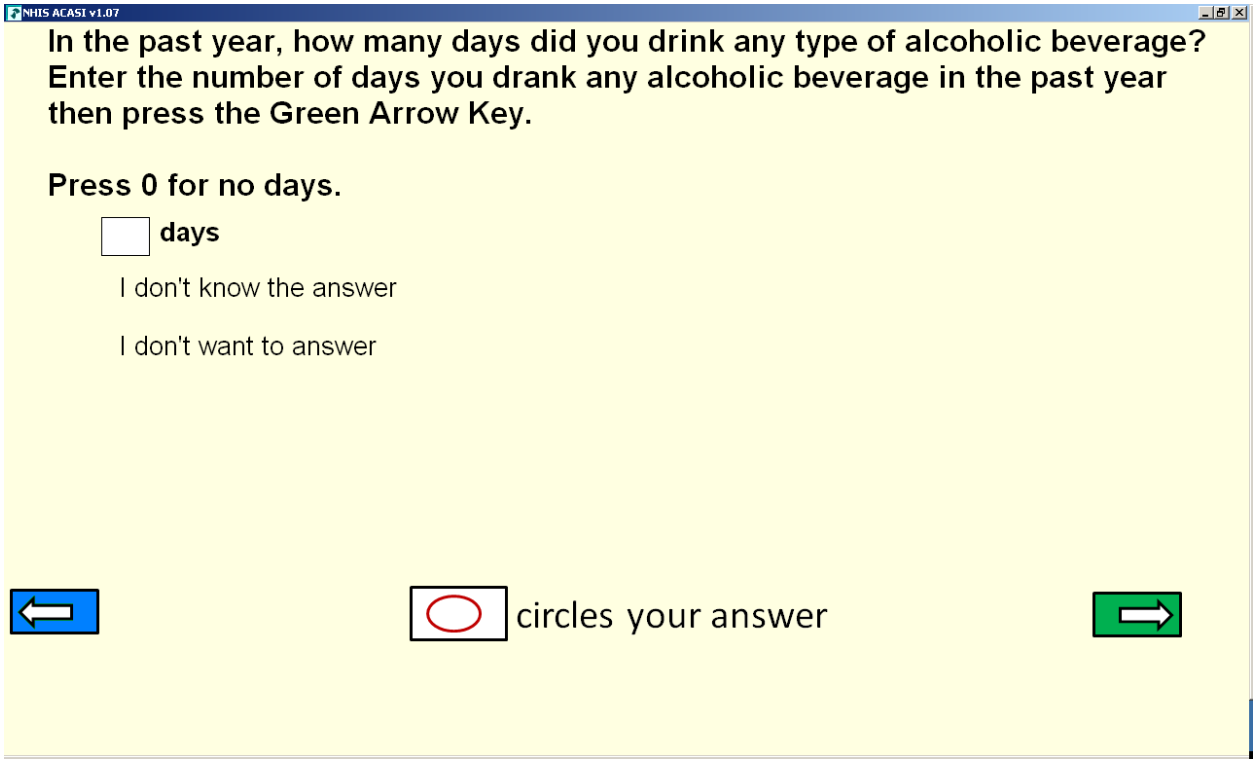

Figure 5 - A Question with an Input Box

3.3 Implementing Questions in the ACASI Module

As stated above, each question and each response option in a question's answer list are defined as separate Fields in the AUXFIELD section of the ACASI module, and their Field descriptions contain Basil question sections that provide the information needed to display the question or the response option.

The authors implemented every question in the NHIS ACASI module using basically the same technique, and they used slight variations on this approach to achieve the desired results for each question. This section of the paper reviews the general technique the authors used to implement a question. It uses the ACIALCDS alcohol use question (shown in Figure 5) as an example.

3.3.1 Basil Question Section for the ACIALCDS Alcohol Use Question

The Basil question section for the ACIALCDS alcohol use question appears below in Figure 6.

```
ACIALCDS_Question
   "<question>
       <audio name=acialcds_q src='ACIALCDS_Q.1.mp3'>
       <label left=45 width=1200 text='<font size=24><B>In the past year, how many days did you
       drink any type of alcoholic beverage?<br>Enter the number of days you drank any alcoholic
       beverage in the past year then press the Green Arrow Key.<br><br>Press 0 for no days.'>
       <input left=100 top=200 width=50 height=40 columns=1
       onenter='blaise:playmedia(acialcds_q)'
       onexit='Blaise:stopmedia(acialcds_q)'
       onreturn='runtime_ACASI_procedures.ACIALCDS_RangeCheck;blaise:save();runtime_ACASI_procedures.ACIALCDS_Check;blaise:nextpage()'
       onkey_1='blaise:gotofield(ACIALCDS_DontKnow)'
       onkey_2='blaise:save();runtime_ACASI_procedures.Special_backup_8'>
       <label left=160 top=200 width=100 text='<font size=20><B>days'>
    </question>" : INTEGER[3], EMPTY
```

Figure 6 - Question Section for Alcohol Use Question

ACIALCDS_Question is defined as an Integer Field with a length of three, and like all questions in the ACASI module it allows Empty.

The audio element in the ACIALCDS_Question question section specifies the identifier (i.e., acialcds_q) by which this question section references the audio file that is to be played. It also

associates a specific recording (i.e., ACIALCDS_Q.1.mp3) with the identifier. (Note that multiple audio files can be specified in the audio element if needed.)

The first label element that appears in this question section specifies the position of the label, the text characteristics, and question text for the alcohol use question.

The input element is a bit more complicated. It first specifies the position and size of the input field. It then lists several control attributes and specifies what actions Basil takes when these events occur.

- When entering the field (i.e., onenter control attribute), play the specified audio file.
- When leaving the field (i.e., onexit control attribute), stop playing the audio recording.
- When the Enter key is pressed (i.e., onreturn), several actions take place.
    - The ACIALCDS_RangeCheck procedure performs a range check,
    - the blaise:save() action saves the current record,
    - the ACIALCDS_Check procedure saves special values in the ACIALCDS storage Field if the focus was on either the Dontknow or Refusal fields when the respondent pressed the Enter key.
    - the blaise:nextpage() action moves the respondent to the next screen on route.
- When the Space Bar is pressed (i.e., onkey_1), the blaise:gotofield action moves the focus from the current field down to the ACIALCDS_DontKnow field which is the "I don't know the answer" option that appears just below the question text on the screen.
- When the Tab key is pressed (i.e., onkey_2), the blaise:save() action saves the current record, and the Special_backup_8 procedure determines what the previous screen was and loads it. That is, the procedure takes the respondent back to the previous screen on route.

The second label element that appears in this question section specifies the position of the label, the text characteristics, and the label ‘days’ for the input box.

3.3.2 Basil Question Sections for the ACIALCDS Alcohol Use Answer List

The Basil question section for the ACIALCDS Dontknow answer option appears in Figure 7.

```
ACIALCDS_DontKnow
   "<question>
       <audio name=dontknow src='DONTKNOW.mp3'>
       <input left=100 width=310 height=30 top=15 columns=1
       AUTOSELECT=false  SHOWSELECTCONTROL=false SHOWFOCUSRECT=false
       onenter='blaise:playmedia(dontknow)'
       onexit='Blaise:stopmedia(dontknow)'
       onreturn='blaise:save();runtime_ACASI_procedures.ACIALCDS_Check;blaise:nextpage()'
       onkey_1='blaise:gotofield(ACIALCDS_Refuse)'
       onkey_2='blaise:save();runtime_ACASI_procedures.Special_backup_8'>
       <ellipse left=45 width=460 height=60 visible=true transparent=yes FocusedBorderColor=red>
    </question> " : (Dont_Know (1) "I don't know the answer")
```

Figure 7 - Question Sections for the Don’t Know Answer Option

ACIALCDS_DontKnow is the first of two options in an answer list for ACIALCDS_Question.

The audio element in the ACIALCDS_DontKnow question section specifies the identifier (i.e., dontknow) by which this question section references the audio file to be played, and it associates a specific recording (i.e., DONTKNOW.mp3) with the identifier.

The input element specifies the position and size of the field. It then lists several events and specifies what actions Basil takes when the events occur.

- When entering the field (i.e., onenter control attribute), play the specified audio file.
- When leaving the field (i.e., onexit control attribute), stop playing the audio recording.
- When the Enter key is pressed (i.e., onreturn), several actions take place.

    - the blaise:save() action saves the current record,
    - the ACIALCDS_Check procedure saves special values in the ACIALCDS storage Field if the focus was on either the Dontknow or Refusal fields when the respondent pressed the Enter key.
    - the blaise:nextpage() action moves the respondent to the next screen on route.
- When the Space Bar is pressed (i.e., onkey_1), the blaise:gotofield action moves the focus from the current field down to the ACIALCDS_Refuse field which is the "I don't want to answer" option that appears just below the "I don't know the answer" answer option on the screen.
- When the Tab key is pressed (i.e., onkey_2), the blaise:save() action saves the current record, and the Special_backup_8 procedure determines what the previous screen was and loads it taking respondent back to the previous screen on route.

The ellipse element that appears in this question section specifies the position of the ellipse and its attributes when the focus is on this Field.

Following the question section is the Field's type declaration. In this case, the ACIALCDS_DontKnow Field is an enumerated type with just one category identifier defined in its list of items. The category text “I don't know the answer” gets displayed on the page.

The Basil question section for the ACIALCDS Refusal answer option appears in Figure 8.

```
ACIALCDS_Refuse
   "<question>
       <audio name=dontwant src='DONTWANT.mp3'>
       <input left=100 width=300 height=30 top=15 columns=1
       AUTOSELECT=false  SHOWSELECTCONTROL=false SHOWFOCUSRECT=false
       onenter='blaise:playmedia(dontwant)'
       onexit='Blaise:stopmedia(dontwant)'
       onreturn='blaise:save();runtime_ACASI_procedures.ACIALCDS_Check;blaise:nextpage()'
       onkey_1='blaise:gotofield(ACIALCDS_Question)'
       onkey_2='blaise:save();runtime_ACASI_procedures.Special_backup_8'>
       <ellipse left=45 width=460 height=60 visible=true transparent=yes FocusedBorderColor=red>
    </question> " : (Refused (2) "I don't want to answer")
```

Figure 8 - Question Section for Refusal Answer Option

ACIALCDS_Refuse is the second of the two options in the answer list for ACIALCDS_Question.

The audio element in the ACIALCDS_Refuse question section specifies the identifier (i.e., dontwant) by which this question section references the audio file to be played, and it associates a specific recording (i.e., DONTWANT.mp3) with the identifier.

The input element specifies the position and size of the field. It then lists several events and specifies what actions Basil takes when the events occur.

- When entering the field (i.e., onenter control attribute), play the specified audio file.
- When leaving the field (i.e., onexit control attribute), stop playing the audio recording.
- When the Enter key is pressed (i.e., onreturn), several actions take place.
    - the blaise:save() action saves the current record,
    - the ACIALCDS_Check procedure saves special values in the ACIALCDS storage Field if the focus was on either the Dontknow or Refusal fields when the respondent pressed the Enter key.
    - the blaise:nextpage() action moves the respondent to the next screen on route.
- When the Space Bar is pressed (i.e., onkey_1), the blaise:gotofield action moves the focus from the current field back to ACIALCDS_Question so the respondent can hear the audio for

the ACIALCDS_Question question again. This action and the playing of a different audio file are the major differences between this Field and the ACIALCDS_DontKnow Field.

- When the Tab key is pressed (i.e., onkey_2), the blaise:save() action saves the current record, the Special_backup_8 procedure determines what the previous screen was and loads it taking respondent back to the previous screen on route.

The ellipse element that appears in this question section specifies the position of the ellipse and its attributes when the focus is on this Field.

Following the question section is the Field's type declaration. The ACIALCDS_Refuse Field is an enumerated type with one category identifier in its list of items. The category text “I don't want to answer” is displayed on the page.

3.3.3 Basil Question Section for the Image on the ACIALCDS Screen

The ACIALCDS_Instructions question section in Figure 9 simply specifies that the image in the Instructions4.bmp file should be displayed at the bottom of the page.

```
ACIALCDS_Instructions
   "<question>
      <label top=180 text='<img src='Instructions4.bmp' transparent=true width=100 height=100 stretch=true>'>
    </question>" : STRING[1], EMPTY
```

Figure 9 - Question Section for Image

4 Integrating the ACASI Module with the NHIS Instrument

The NHIS instrument consists of four main components. These are the Demographic, Family, Sample Child, and Sample Adult questionnaires. The ACASI module was added to the end of the Sample Adult questionnaire. This section of the paper briefly describes how the authors integrated the ACASI module into the instrument.

4.1 Modifications to the Blaise Instrument

To add the ACASI module to the NHIS Blaise instrument, the authors created a new block (named ACI) and included it in the Sample Adult questionnaire.

The ACI block defines Fields that correspond to the data Fields collected in the ACASI module. The data collected in the ACASI module is passed back to the Blaise instrument and stored in these Fields. As a result, the data captured in the ACASI module is saved in the NHIS instrument’s Blaise database, and all the data collected during an NHIS interview is available in one place for the sponsor.

The ACI block also defines two introductory screens that the NHIS instrument displays when it is time for the FR to transfer the laptop to the respondent for the administration of the ACASI module. The first screen prompts the FR to describe ACASI to the respondent, and the second screen provides FR instructions for transitioning the laptop to the respondent. These screens appear below in Figures 10 and 11 respectively.

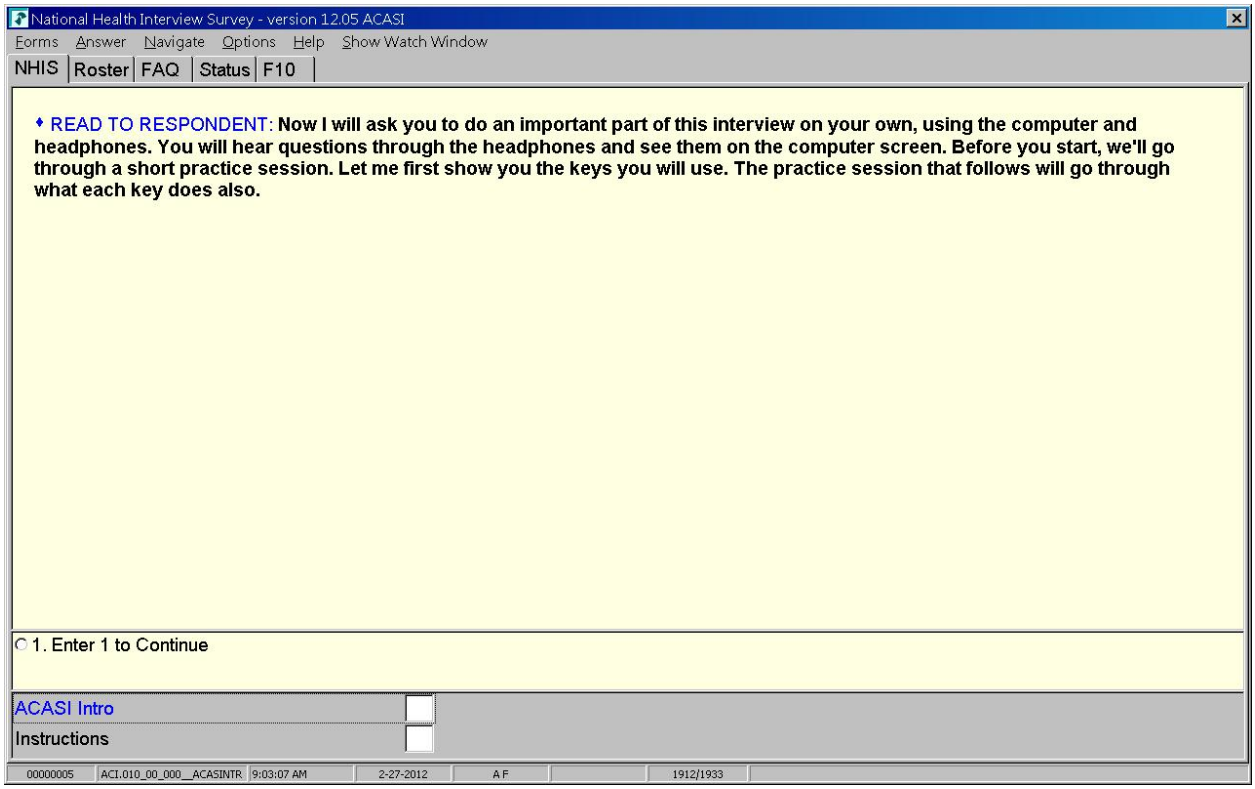

Figure 10 - Explaining ACASI to the Respondent

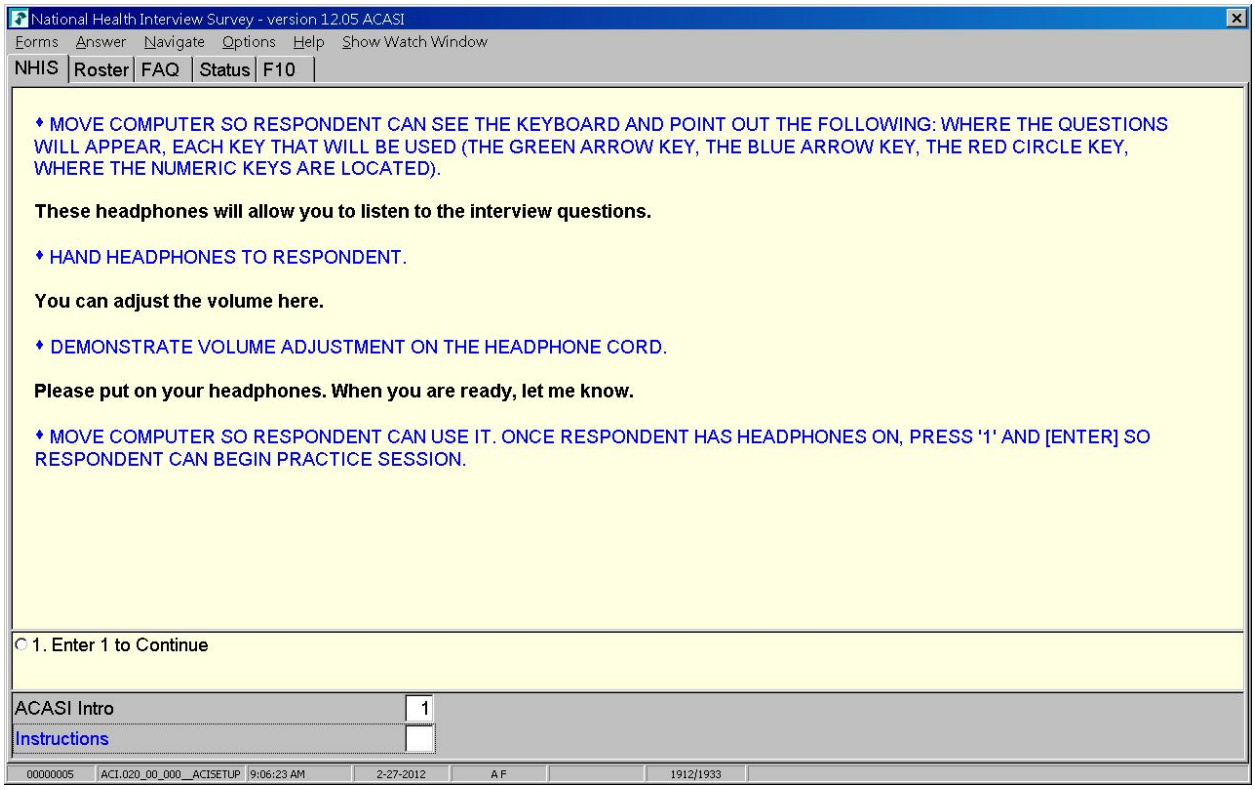

Figure 11 - Transitioning the Laptop to the Respondent

After administering the introductory screens, the ACI block calls the procedure prc_launchACASI. This procedure in turn calls the alien procedure script_launchACASI to launch Basil.exe and run the ACASI module.

4.2 The script_launchACASI Alien Procedure

The first thing the script_launchACASI alien procedure does is create and open a new database for the Basil ACASI module (i.e., NHIS_ACASI.bdb). The procedure then transfers the values of all the Fields (defined in the Adult.ACI block) from their storage locations in the Blaise database into the corresponding Fields in the Basil database. The Field must exist (with the same name) in both the Blaise instrument and the Basil instrument, and AUXFIELD values are not transferred.

The complete transfer of the Fields from the Adult.ACI block in the Blaise instrument's database to the Basil database might seem a bit excessive in light of the following facts.

- Only a few of these Fields (e.g., SEX) are populated at this point in time.
- Neither the FR nor the respondent will be allowed to re-enter the ACASI module once the respondent competes it.

However, it was a simple feature to implement, and if the requirement to lock the users out of this module is ever removed in the future, then the functionality for repopulating the Fields in the Basil instrument's database is already in place.

After the Field values are transferred from the Blaise database to the Basil database, the procedure calls the Basil.exe and passes the name of the Basil database and its primary key (CASEID) as parameters. The WAIT argument is used to suspend execution of the procedure until the new process (i.e., Basil.exe) terminates.

When the Basil.exe closes and control returns to the alien procedure, the procedure transfers the values of all Fields from the Basil database to the corresponding storage locations in the Blaise database (in this case the block Adult.ACI). Again, the Field must exist (with the same name) in both the Basil instrument and the Blaise instrument, and Auxfield values are not transferred.

The procedure turns on a flag (i.e., ACASIFLG) to indicate that the ACASI section is complete. This is done so that the rules in the Blaise instrument route to a "lock screen" to prevent the FR from re-entering the Basil module.

Finally, the procedure deletes the Basil database since it is complete and will not be accessed again by either the respondent or the FR.

5 Issues Developing and Deploying the ACASI Module

There were a number of issues that arose during the design and development of the ACASI module for the 50 case test. This section discusses some of the more noteworthy issues.

5.1 Navigation in the ACASI Module

Getting the navigation in the ACASI module to work correctly was problematic at times. Typically, the module worked as expected when moving straight through it without backing up. However, irregular navigation (such as repeatedly backing up and moving forward or backing up, changing an answer, and moving forward) would sometimes cause a problem. For example, the focus might not fall on the correct field on the screen or the instrument might completely freeze.

In general, we found that these problems were the result of a conflict between the rules and a task performed by an event handler we had programmed. Ultimately, the authors reduced the rules section of the ACASI module to the bare minimum and controlled most of the navigation in the module with event handlers.

5.2 Off Path Data

Unlike Blaise, Basil does not handle data that has been backed over and then placed off route. Backed over data are retained in a Basil application when it terminates.

While it is possible to program a script to clean up the off path data when the ACASI module terminates, this is essentially performing a data processing task in the field. This is an activity that the Census Bureau's authors prefer to avoid. After discussing the issue with the subject matter specialists in the DSD and the NCHS, all parties agreed to leave the off path data alone. The sponsor's data processing staff would cull the off path data after it came in from the field.

The ACASI module is small, and there are only a couple of places in the module where the respondent can back up, change an answer, and move forward down a new path resulting in off path data. As a result, the problem seemed to be minor and easily handled by the data processors.

5.3 Time between Screens

The authors encountered an issue with the time it took to move from one screen in the ACASI module to the next. For the most part this was limited to one screen that took 5 to 10 seconds before the audio played and the text displayed. This particular lag in the ACASI module was consistent on all the desktop workstations that were used to test the ACASI module. One developer actually had a significant lag between all the screens in the module when he tested it on his workstation. His machine was an older model and probably an exception.

The authors have not yet determined the cause of the problem. Researching the problem became less of a priority when they discovered that the problem did not occur when running the ACASI module on the FR laptops.

There is a significant difference between the workstations used by the authors for development and the FR laptops. For example, the operating systems are different. The developers' desktops run Windows XP while the laptops run Windows Vista. There are probably differences in the Federal Desktop Core Configuration (FDCC) settings as well. Some recent research suggests that the problem has something to do with the .mp3 audio file used for the screen experiencing the delay.

5.4 Transmitting the Instrument to the FR Laptop for the 50 Case Test

The NHIS ACASI module required files that we do not usually deploy with our instruments. Obviously audio files were required for the ACASI module. These were created in .mp3 format. The module also used image files to provide visual clues to the respondent on how to use the instrument. These were bitmap files (.bmp).

Table 1 provides the type, number, and total uncompressed size of the additional files required for the ACASI instrument.

Table 1. ACASI Specific Files Required by the NHIS Instrument

File Type	Number of Files	Total Size in Bytes
.mp3	162	9,040,750
.bmp	8	1,456,024
Totals	170	10,496,774

These files quadrupled the size of the typical NHIS instrument "package" that we transmit to all NHIS FRs. Whenever we have large amounts of data to transmit to FRs, we always try to find an approach that has the least impact on the users. Since the 50 case test involved only 6 users, we decided to customize our transmission scripts so the additional files were sent separately from the instrument package and were sent only to the 6 FRs participating in the test. While all the NHIS FRs received the 50 case test instrument, only the 6 FRs working the test received the audio and image files. As we move into the larger tests or into production, all NHIS FRs will have to receive these files (which could possibly change with each instrument release) therefore increasing the length of time our quarterly NHIS instrument transmissions will take.

5.5 Testing the ACASI Module

Testing an ACASI application can be time consuming. It is necessary to listen to every recording completely, and there can be quite a few recordings in an application. For example, in the NHIS ACASI module, a simple question like the one that appears below requires six audio files.

Were any of your grandparents born in the United States?
Yes
No
I don't know the answer
I don't want to answer

When the respondent lands on the question, two audio files play. The first one states the question and the second one reads the complete answer list. Then there are individual recordings for each of the response options in the answer list. These play as the focus moves from one answer to the next when the respondent uses the Space Bar to cycle through the answer list. Consequently, there are a total of six audio files that must be played in their entirety to test this question. This is a relatively simple question. Many of the questions in the ACASI module had much longer answer lists and therefore more recordings to test.

Testing an ACASI instrument that administers the questionnaire in a language that you do not speak is another challenge. In this situation, it is useful to have a copy of the specification with the question text in the foreign language and a flow chart of the instrument.

As with any automated questionnaire, it is necessary to test irregular navigation. Testing the ability to back up and move forward and back up and change paths in the instrument seems to even more critical in an ACASI application developed in Basil.

6 Future Initiatives

At the time this paper was written, the Census Bureau was completing work on the ACASI module for the 1,100 case test. The Census Bureau is scheduled to field test this instrument in April 2012. Several new features were added to the ACASI module for this test. Plans for the 11,000 case test were also being prepared. Some of the initiatives for those two tests are presented here.

6.1 Respondent Debriefing Questions

Eight respondent debriefing questions were added to the end of the ACASI module for the 1,100 case test. These questions ask respondents for feedback on their ACASI experience. The debriefing questions are listed below.

- Did the interviewer explain how to use this computer?
- Did you have any difficulty using the keys to choose your answers?
- Did you listen to the recorded voice for all of the questions, some of the questions, or none of the questions?
- Was the recorded voice too slow, to fast, or about the right speed?
- Was the recorded voice to low, too loud, of about the right volume?
- Did you use headphones or earbuds?
- How comfortable were the headphones/earbuds?
- Do you think it was important for this part of the interview to be private?

Many of these questions were included in the respondent debriefing based on observations made during the 50 case test. The NCHS wanted to get better feedback from the respondents about their ACASI experience.

6.1.1 Audio Recordings

After the 50 case test, there were some comments that the narrator in the (English) audio files was speaking too slowly. Respondents who listened to all of the recordings that came on route (in their entirety) took a considerable amount of time to get through the ACASI module. Several respondents took over 10 minutes to complete the ACASI questionnaire, and one respondent took 20 minutes to complete it. The survey methodologists plan to review the respondent debriefing data from the 1,100 case test to determine whether the audio files need to be rerecorded.

6.1.2 Audio Equipment

The 50 case test used headphones with disposable covers. The FRs sometimes had difficulty putting the covers on the headphones. This caused a delay in transitioning the laptop from the FR to the respondent, and in some cases seemed to cause some concern on the part of the respondent. One proposed solution to this problem was to use larger covers if they were available. Another suggestion was to offer the respondents a choice of the headphones or earbuds that they could keep or throw away. For the 1,100 case test, respondents will be given a choice between the headphones and the earbuds. Again, the survey methodologists will review the respondent debriefing data determine what equipment works best.

One participant in the 50 case test did not want to use the headphones. Others wanted to use their own headsets. The FRs had not been trained to handle this situation so they had issues getting the headsets to work in some cases. Sometimes the FR could not get the respondent's headset to work with the FR laptop. Headsets with an analog audio jack worked. Headsets with a USB connection might or might not work depending on whether the laptop had the appropriate device driver installed.

6.2 Audit Trails

The ACASI module used during the 50 case test lacked audit trails because there had not been enough time to implement this feature before the test. There will be audit trails for the 1,100 case test

The author implemented the audit trail functionality using Maniplus procedures. (Much of this code was borrowed from a Blaise example.) There were procedures for initializing a temporary audit trail file, formatting the audit trail records and writing them to the temporary file, appending the records in the temporary file to the Blaise instrument's audit trail file, and closing out the audit trail process when the respondent exits the ACASI module.

The initialization and close out procedures are called from the oncreate and onclosequery event handlers (respectively) that are present in the Basil module's application section.

The procedure that formats and writes an audit trail record (WriteAuditTrail) is called from within each of the module's question sections under several conditions. The procedure is called:

- when entering a field (onenter),
- when the respondent presses the Enter key to save an answer and move to the next screen (onreturn),
- when the respondent presses the Space Bar to scroll to the next option in an answer list (onkey_1), and
- when the respondent presses the Tab key to back up to the previous screen (onkey_2). (The onkey_1 and onkey_2 events are defined in the module's application section.)

When an event handler in a question section calls the WriteAuditTrail procedure, it passes a string to the procedure that identifies the question from which it was called and the type of event that occurred. This allows the WriteAuditTrail procedure to write out an audit trail record with the appropriate information.

In order to write to the Blaise instrument's audit trail file from the Basil application, it was necessary to modify the audit trail information file (.aif) so the CloseFile option is set to 1. This instructs the Blaise Audit Trail dll to open and close the audit trail file every time it writes to the file. Since the

audit trail file is closed when the ACASI module is running, the audit trail file is available for the ACASI module to write to it.

An example of the ACASI audit trail appears below. In this example, the first seven lines are from the Blasie NHIS instrument and the rest from the ACASI modules.

```
"2/28/2012 3:37:40 PM","Enter Field:ACI.ACILANG","Status:Normal","Value:"
"2/28/2012 3:37:42 PM","<KEY:>21[ENTR]"
"2/28/2012 3:37:45 PM","Action:Store Field Data","Field:ACI.ACILANG"
"2/28/2012 3:37:45 PM","Leave Field:ACI.ACILANG","Cause:Next Field","Status:Normal","Value:1"
"2/28/2012 3:37:45 PM","Enter Field:ACI.ACASINTR","Status:Normal","Value:"
"2/28/2012 3:37:46 PM","<KEY:>1[ENTR]"
"2/28/2012 3:37:47 PM","Action:Store Field Data","Field:ACI.ACASINTR"
$2/28/2012  3:37:51 PM                                                                                    999
$2/28/2012  3:37:52 PM  ENTER FIELD  ACIHEAD
$2/28/2012  3:43:09 PM  LEAVE FIELD  ACIHEAD                GREEN ARROW
$2/28/2012  3:43:09 PM  ENTER FIELD  ACIINTR1
$2/28/2012  3:43:30 PM  LEAVE FIELD  ACIINTR1               GREEN ARROW
$2/28/2012  3:43:30 PM  ENTER FIELD  ACIINTR2
$2/28/2012  3:43:49 PM  LEAVE FIELD  ACIINTR2               GREEN ARROW
$2/28/2012  3:43:49 PM  ENTER FIELD  ACIDLIC_Question                   ACIDLIC
$2/28/2012  3:44:09 PM  LEAVE FIELD  ACIDLIC_Question       RED CIRCLE  ACIDLIC
$2/28/2012  3:44:09 PM  ENTER FIELD  ACIDLIC_Yes                        ACIDLIC
$2/28/2012  3:44:11 PM  LEAVE FIELD  ACIDLIC_Yes            RED CIRCLE  ACIDLIC
$2/28/2012  3:44:11 PM  ENTER FIELD  ACIDLIC_No                         ACIDLIC
$2/28/2012  3:44:13 PM  LEAVE FIELD  ACIDLIC_No             RED CIRCLE  ACIDLIC
$2/28/2012  3:44:13 PM  ENTER FIELD  ACIDLIC_Question                   ACIDLIC
$2/28/2012  3:44:16 PM  LEAVE FIELD  ACIDLIC_Question       RED CIRCLE  ACIDLIC
$2/28/2012  3:44:16 PM  ENTER FIELD  ACIDLIC_Yes                        ACIDLIC
$2/28/2012  3:44:19 PM  LEAVE FIELD  ACIDLIC_Yes            GREEN ARROW ACIDLIC   1  Yes
$2/28/2012  3:44:19 PM  ENTER FIELD  ACINUM_Question                    ACINUM
$2/28/2012  4:03:33 PM  LEAVE FIELD  ACINUM_Question        GREEN ARROW ACINUM
```

Figure 12 - ACASI Audit Trail

Note that the ACASI audit trail records differ significantly from the Blaise audit trail records. The format of the ACASI audit trail records was the Blaise author's attempt to make the ACASI section of the audit trail as readable as possible. While this approach is creative, it has a downside. Both the Census Bureau and the NCHS use applications that parse the audit trail files to obtain timing information and other data. These applications will not be able to process the ACASI audit trail records without some modifications. Since the NCHS uses an application developed by another organization, modifying their application is not an option. For the 1,100 case test the ACASI audit trail format will remain as it is. The DSD data processors will strip the ACASI records (all of which begin with a "$" to make identifying them easy) from the audit trail before passing them along to NCHS. After the 1,100 case test, the authors will need to modify the ACASI audit trail procedure so it writes out records that have the same format as the Blaise audit trail. The resulting ACASI audit trail might be less readable, but it will conform to all the systems that use the audit trail files.

6.3 Spanish

The ACASI module for the 1,100 case test will have a Spanish component. Unfortunately, Basil does not support multiple languages the way Blaise does. The NCHS programmer's solution for the multiple language issue was to create two separate datamodels – one in English and the other in Spanish. The Census Bureau's authors had hoped to find a better solution, but ended up settling on a very similar approach. The Census Bureau's authors created a single datamodel in which there were two paths – one for English and one for Spanish. This approach is essentially the same as the NCHS method. Neither technique is satisfactory. Both complicate the programming and testing of the ACASI module. The authors have to keep the logic of the two paths in synch, and the testing effort doubles because there are two unique paths to test. The Census Bureau's Authoring Staff continues to look for a better way to implement multiple languages in Basil.

The issue with the lag between screens in our development environment got worse when the authors added the Spanish to the ACASI module. The lag at the one screen in the English path remained the same, but several screens in the Spanish path had similar delays before the audio played and the question text appeared on the screen. Like the lag on the English path, this was a problem on the developers' workstations only. The Spanish component of the ACASI questionnaire ran without the delays on the FR laptops.

Adding Spanish to the ACASI module essentially doubled the total size of the audio files needed to administer the questionnaire. This further aggravates the issue of transmitting the instrument to the FR laptops.

6.4 Edit Checks

Edit checks for numeric fields were not available in the 50 case test instrument. The authors had programmed the edit checks, but they were in a version of the NHIS instrument that had not been thoroughly tested. As a result, an earlier version of the instrument without the edit checks was used for the test.

The 1,100 case test instrument will have edit checks for the numeric fields. These are implemented using Maniplus procedures and dialogboxes. The figure below displays an example of one of the error messages.

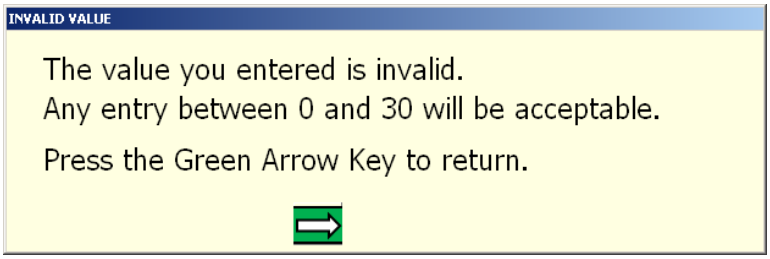

Figure 13 - Range Check Dialog Box

While these edit checks are adequate, the dialogbox does not play audio that corresponds to the text. The edit checks that use the Maniplus dialogboxes will be replaced for the 11,000 case test. When respondents enter a value that is out of range in the 11,000 case test, the ACASI module will direct them to a new screen in the instrument that will inform them visually and acoustically that there was an error. The ACASI module will then return them to the previous screen where the respondents can make the correction.

7 Conclusion

There are some advantages and disadvantages in using Basil to develop an ACASI questionnaire. Basil's two big advantages are that it gives the developer considerable control over the appearance of an application's screens, and it also gives the developer control over actions that need to take place upon the occurrence of certain events. When compared to Blaise, Basil's main disadvantage is that it is not as easy to program. The Basil documentation that comes with the Blaise Control Centre states, "By default your BASIL application does not do too much." This is true. The programmer must manage a number of issues that Blaise would handle automatically or could be implemented in Blaise with less effort than in Basil. This paper addressed some of these issues such as the support of multiple languages, irregular navigation, handling data that is off route, writing an audit trail, and so on.

An ACASI module would have been easier to implement and maintain in Blaise, and it would have been more robust and less problematic in the field. However, a Blaise ACASI module would not have provided the simple user interface that the customer requested.

While the ACASI pilot project is progressing satisfactorily, the project has not been problem free. An initial lack of experience with Basil at both the NCHS and the Census Bureau was an issue. The compressed project schedule probably contributed to some of the issues we encountered as well. Rather than taking the prototype developed by the NCHS and simply expanding the application as we did, it might have been useful to take the time to analyze the prototype and determine whether there was a better way to implement it.

Since the NCHS provided the ACASI prototype, it has grown substantially. We would like to find a way to reduce the amount of code and simplify the task of maintaining the ACASI module. The authors at the Census Bureau have started to reevaluate how the Basil module was designed to determine whether there is a better approach. Most questions in the NHIS ACASI module have several question-specific event handlers programmed in Manipula to control some functionality. The programmers are investigating whether they can use a more generic approach that might make use of generalized templates to avoid all the question specific procedures. There are probably other ways to simplify the Basil ACASI module. For example, instead of customizing the ellipse to fit the various sizes of the answer categories, it would be easier to use a generic rounded rectangle that spans the page. Such a rectangle would fit all answer categories, and there would be no need to customize the ellipse for each answer in each question's answer list. (Because this sort of change affects the ACASI module's GUI interface, we would have to negotiate the implementation of such a change with the sponsoring agency.)

8 Acknowledgements

The author would like to acknowledge the following people for input into this paper.

Andrea Piani, U.S. Census Bureau
Andrew Stevenson, U.S. Census Bureau
Ann Daniele, U.S. Census Bureau
Anna B. Sandoval Girón, National Center for Health Statistics
Jason Arata, U.S. Census Bureau
Karen Bagwell, U.S. Census Bureau
Malcolm Robert Wallace, U.S. Census Bureau

9 Disclaimer

The views expressed on (statistical, methodological, technical, or operational) issues are those of the author(s) and not necessarily those of the U. S. Census Bureau.