SRCBlaise Standards

University of Michigan

Table of Contents

Disclaimer

This book is divided into three major sections; Screen Standards, Specification Standards, and Programming Standards. There are also a number of appendices and an index at the back. The Screen Standards are aimed primarily at Blaise Programmers, but can also be used by Project Managers and Spec Writers as they develop their survey specifications and need to have an idea of screen design and how it is applied for various types of questions. The Specification Standards are aimed at Spec Writers and Project Managers as a guide to assist them in producing clear and complete Blaise specifications. The final section, Blaise Programming Standards, is designed primarily for the the CAI Programmers. The information contained within this section is very specific to many of the base standards that all SRC Blaise applications should adhere to.

In Appendix A, there is a very useful "Quick Reference Guide" that shows a variety of question types. The guide is split into four panels per question; the first is an example of the question specified according to the specifications standards, the second is an example of how the question is programmed according to the programming standards, the third is how the question looks in the DEP according to the screen standards, and the fourth panel displays the question in MQDS (the Michigan Questionnaire Documentation System) for documentation purposes.

Thus, it is not necessary for all users of this book to assimilate all pieces of it; if anything, the Blaise Programmers are the part of the intended audience in which all parts of the book are equally important. Everyone else should mainly pay attention to the sections of the book that are most cogent to their particular role on a given project.

Edition Number: 2007_1
Edition Date: October 2007

Edition Number: 2008_1
Edition Date: March 2008

Summary of Major Revisions:

- Changed placement of Field Description in all examples of a question specification
- Added a numbering scheme to Signals and Checks
- Modified an example to show a Field Name and Field Tag that are different
- Added a disclaimer about each section's intended audience
- Added programming guidelines related to the use and re-use of fills
- Added edition numbers and edition release dates
- Added updated screen shots where necessary

Foreword

These SRC Blaise Standards consolidate and make consistent various guidelines developed over several years for specification and programming of Blaise computer assisted interview (CAI) survey instruments, and design of the Blaise CAI interface. To facilitate understanding of the relationship among specification, programming, screen design, and instrument documentation, they also include a reference section showing specifications and program code side-by-side with resulting CAI screens and documentation produced by the Michigan Questionnaire Documentation System (MQDS).

The use of the term "standards" is deliberate, and is meant to convey that SRC expects CAI instruments to be specified and programmed consistently, and that programming instruments according to these standards is not optional. At the same time, it is recognized that these standards are not comprehensive enough to meet the needs of all survey designs.

Thus, in a sense they are guidelines, as are all such standards. Some survey designs may require deviating from them, and they will necessarily evolve over time. It is the obligation of each project team involved in CAI specification and programming to adhere to them, and to help them evolve as needed. To that end the SRO CAI programming team, working closely with SRC program staff and the SRO Project Design and Management Group (PDMG), will initiate periodic review and updating of these standards.

User comments are welcome, and may be sent via email to: Blaise_Stds@isr.umich.edu.

Blaise Screen Design Standards

Blaise Screen Design Guidelines • The Basics

This section provides basic screen design guidelines for Blaise survey instruments, with a primary focus on interviewer-administered surveys. First, the key principles behind the guidelines are listed. These are based on more general guidelines on user-centered design, from research on human-computer interaction. They are followed by standards established for display of text and interviewer instructions on the Blaise screen, and examples of screens that demonstrate application of these standards.

Key principles

- Consistent screen design;
- Visual discrimination among the different elements (so that CAI users learn what is where, and know where to look);
- Adherence to normal reading behavior (i.e., start in upper left corner);
- Display of instructions at points appropriate to associated tasks (e.g., the show card or respondent booklet instruction precedes the question and the entry instruction follows the question text);
- Elimination of clutter and unnecessary information or other display features that distract users from immediate tasks (e.g., lines and toolbars).

Text characteristics

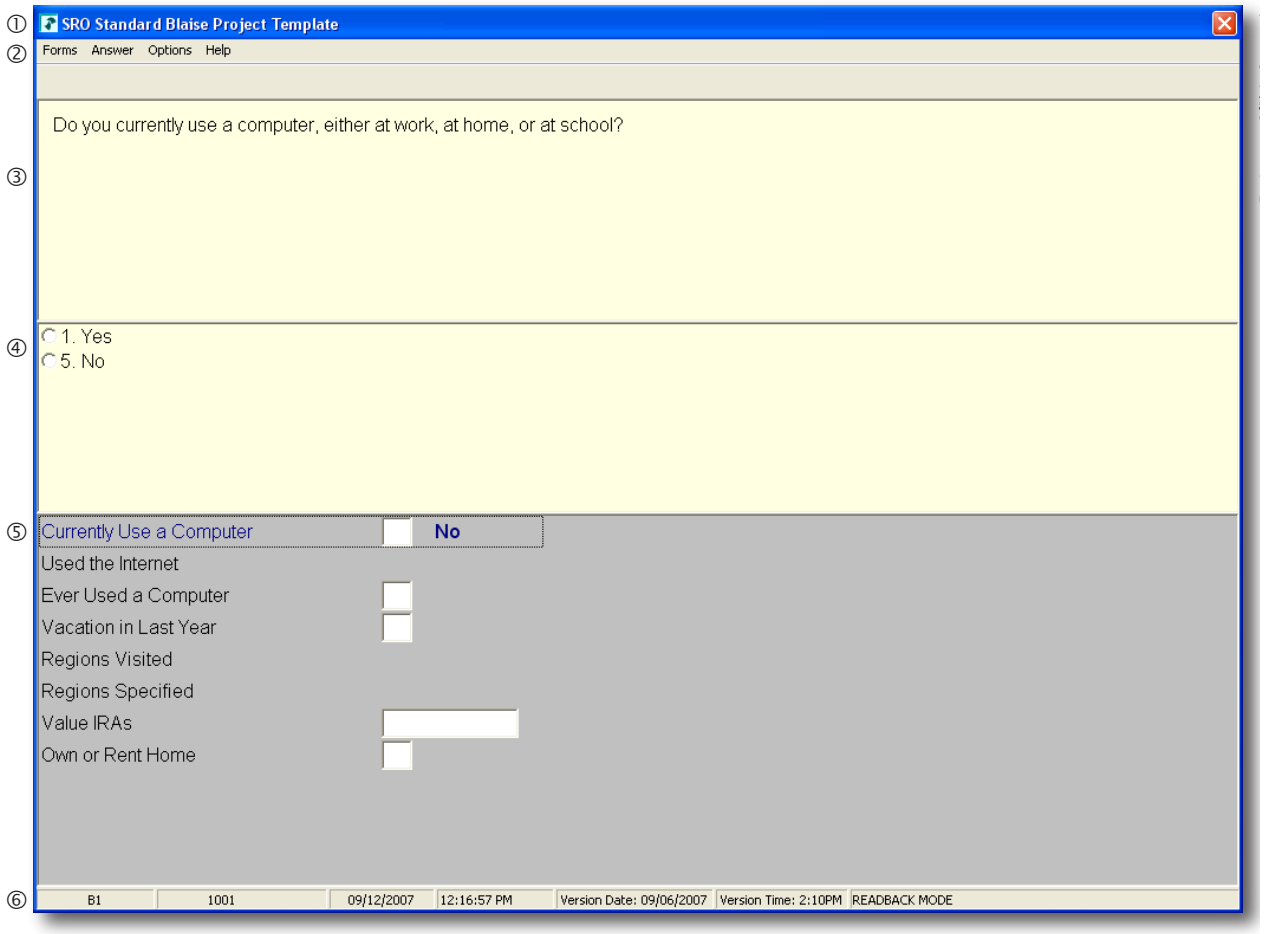
- Display question text on a light background color (cream), in mixed case, and in 12-point Arial, black;
- Display instructions in **11 point Arial bold blue**;
- Display response categories
  - that could be read to the respondent in 12-point Arial black
  - that would not be read to the respondent in **11-point Arial bold blue;**
- Use underline for emphasis, sparingly;
- Place optional text in (parentheses);
- Display in-text references to function keys and numbers to type in mixed case within square brackets, for example, [Enter], [1], [F12], and [Ctrl-R].

On-Screen Instructions and Other Information

- Place references to interviewer aids (e.g., an event history calendar or show card instruction) and the question text in the upper left corner of the screen, above the question text;
- Place instructions that precede the question flush left with the question;
- Use icons to distinguish special instructions:
  - 🕮 **Page 1**, for respondent booklet instruction;
  - 🗏 **Calendar**, for event history calendar instruction, and;
  - ☑ **Interviewer Checkpoint**.
- "Bullet" all other interviewer instructions with an 11-point bold blue diamond (♦);
- Single space within an instruction and double space between instructions;
- Place an online help indicator (**[F1]-Help**) above the question on the right margin, for questions with "question-by-question objectives" (QxQ's);
- Indent instructions that follow the question;
- Place any context-related information below the question-level help indicator on the right margin (for example, changing person-level information as the interviewer navigates a household roster or grid);
- Display instructions in the order associated with required interviewer tasks;
- Include an actual question in explicit interviewer checkpoints, displayed in **11-point Arial bold blue**;
- Capitalize only key task-related action verbs (ASK, READ, ENTER, and PROBE), and only at the beginning of instructions;
- Keep instructions simple and concise;
  - Put long instructions or those not directly related to asking questions or entering responses into online help (QxQ's).
- Conditional instructions start with the conditional phrase, not the action verb, and the action verb is not capitalized (e.g. conditional probes and data entry instructions);
- In probe instructions, place text to be read in Arial black;
- Place references to respondent answers in quotation marks.

Anatomy of the Blaise Screen

There are some fundamental elements that are common to most Blaise CAI instrument screens. The following screen highlights these elements.

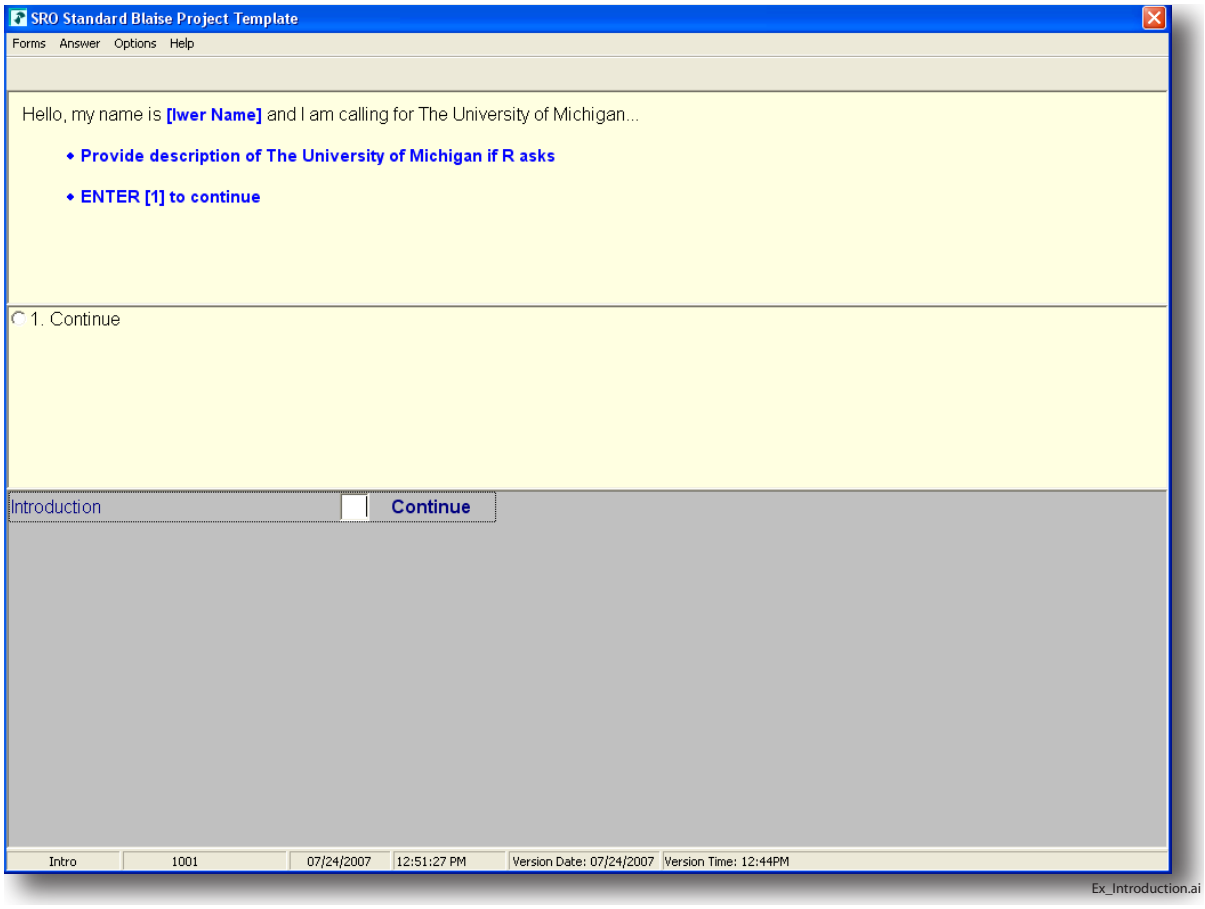


1. The Windows "Title Bar" identifies the current project.
2. The Windows "Menu Bar" contains drop-down menus for various user tasks.
3. The "Question Window" is the portion of the screen that includes the question text and any instructions to the user.
4. The "Response Option Window" contains the response categories for the current question.
5. The "Entry Window" is the section of the screen in which a response is entered. It also displays the Field Description (25-character maximum) and answer name.
6. "Status Bar" from left to right contains the following:
   - Field Name; generally alpha-numeric
   - Sample ID
   - Current Date
   - Current Time
   - Instrument Version Date
   - Instrument Version Time
   - Readback Mode Indicator (when Readback mode is on)

Examples

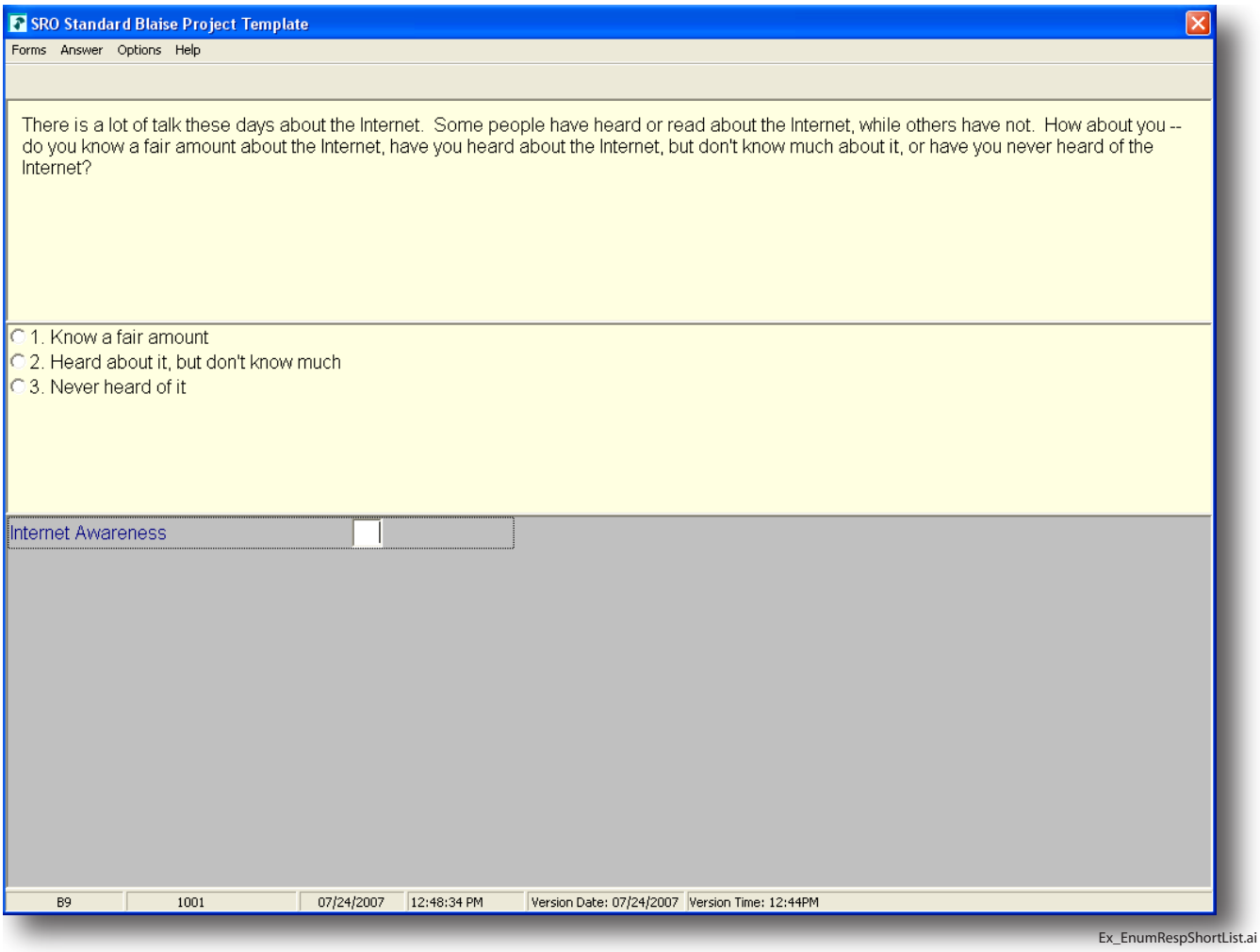
Introduction

- Non-question text to be read to the respondent;
- **ENTER [1] to continue** instruction;
- Bulleted instructions (♦), **11 point Arial bold blue**, in task order, indented after question text;
- Capitalized key task-related action verb, ENTER;



Enumerated Response Options (Short List)

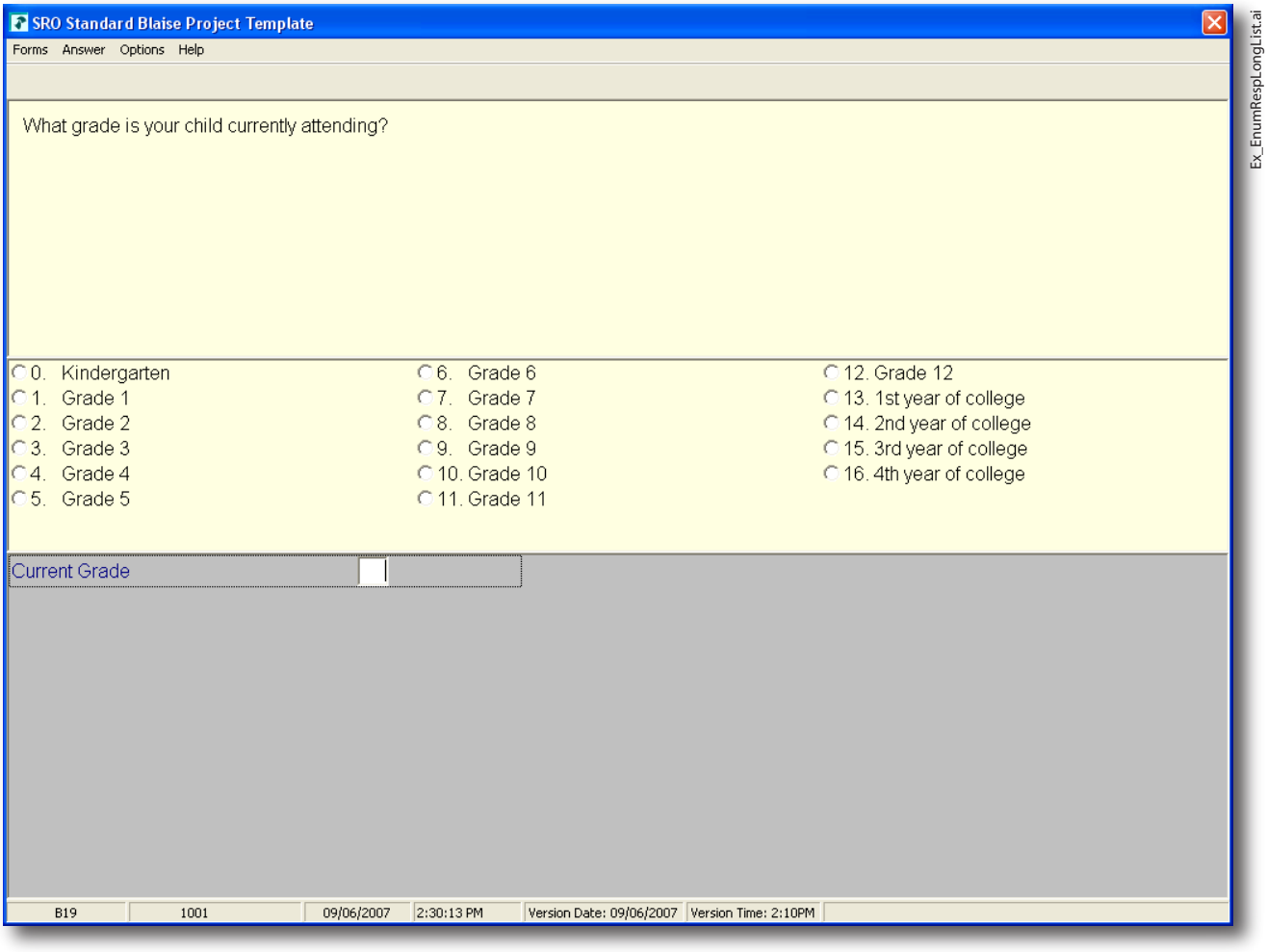
- Less than six response options;
- Single column;
- Numbered sequentially, unless otherwise specified; note that the SRC convention for Yes and No is to number them as 1 and 5;



Ex_EnumRespShortList.ai

Enumerated Response Options (Long List)

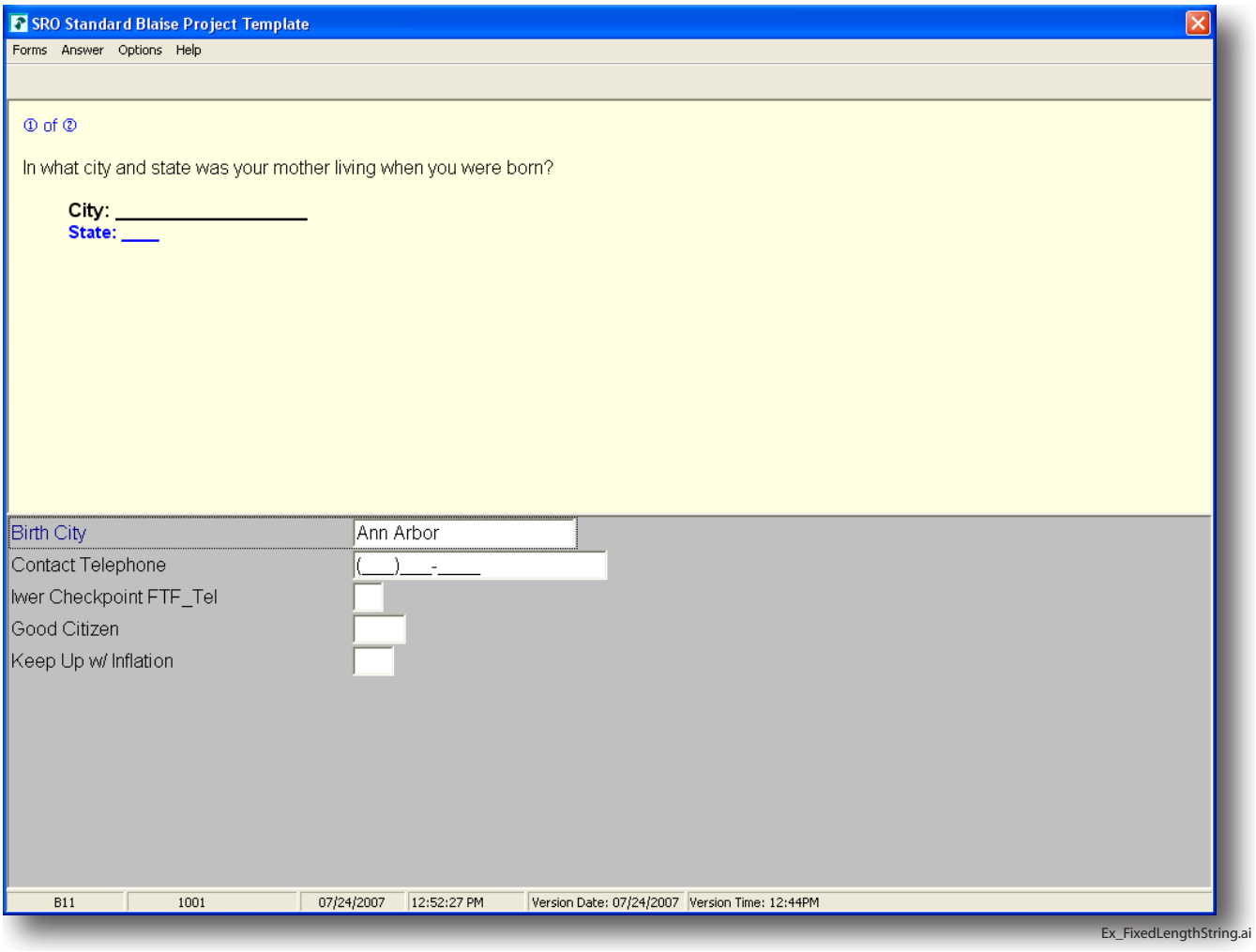
- More than six response options;
- Multiple columns, to avoid scrolling; "balanced" across response option window;
- Numbered sequentially,



Ex_EnumRespLongList.ai

Fixed Length String

- Short open response;
- Single-line, ticker-tape entry field;

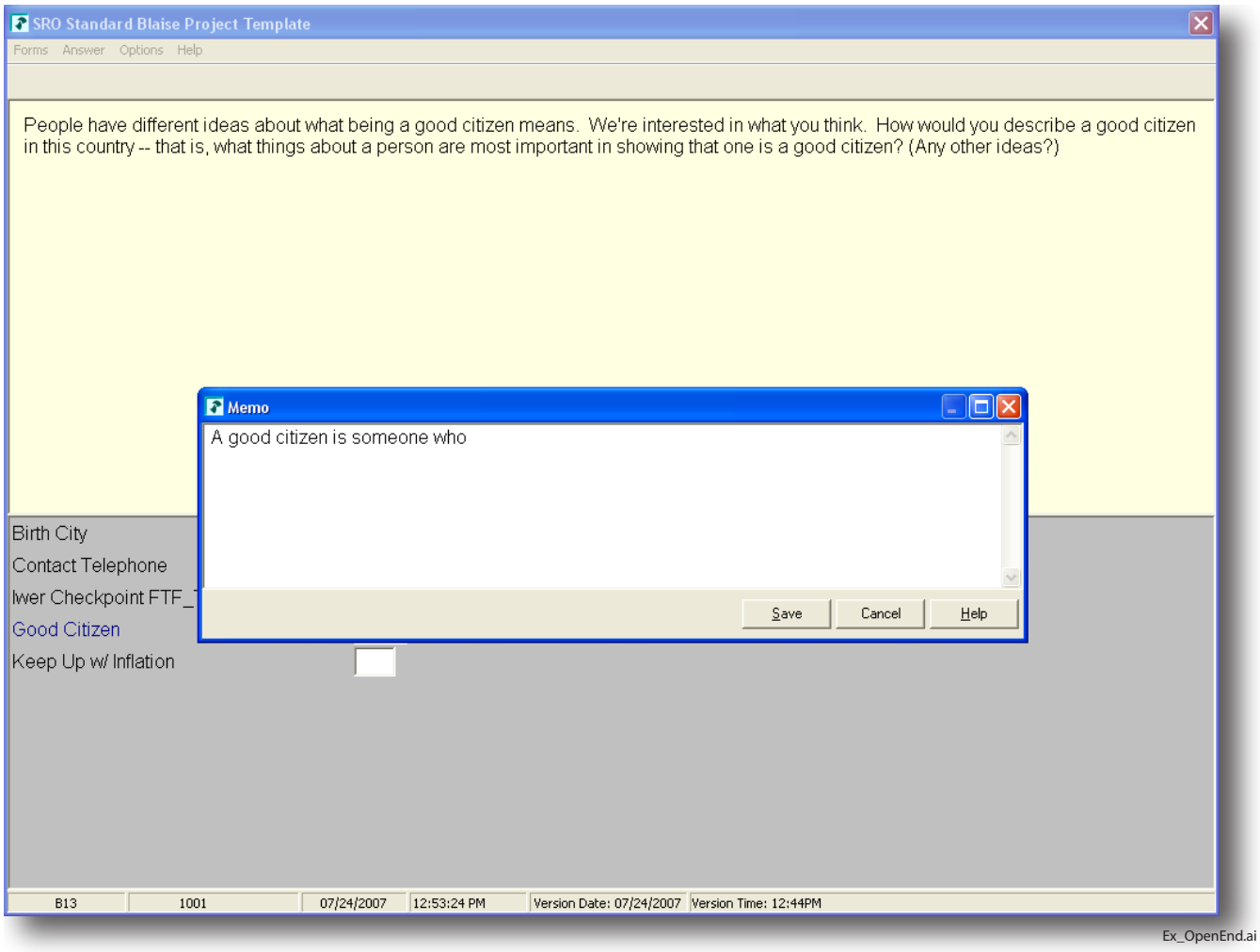


Ex_FixedLengthString.ai

Note that Blaise stores strings in the data file and that codebooks generated from MQDS outputs a 1 (asked) or a 0 (not asked).

Open End

- Long open verbatim response;
- "Free Form" Window for input;
- Press [Insert] to edit existing text;

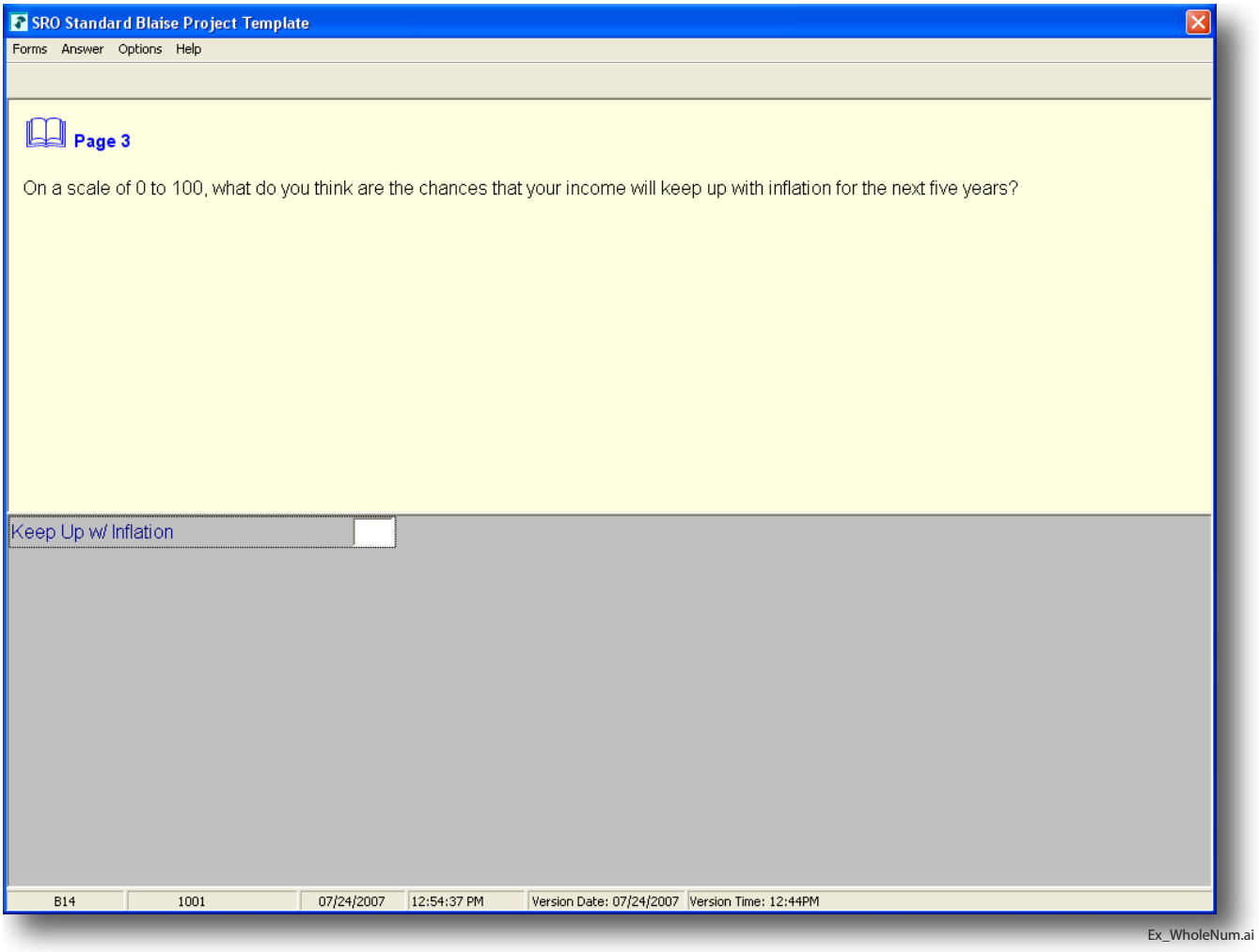


Ex_OpenEnd.ai

Note that Blaise exports open end data to a separate file (.opn)*

Whole Number

- Integer or whole number response;
- Width of input field equals width of maximum valid response

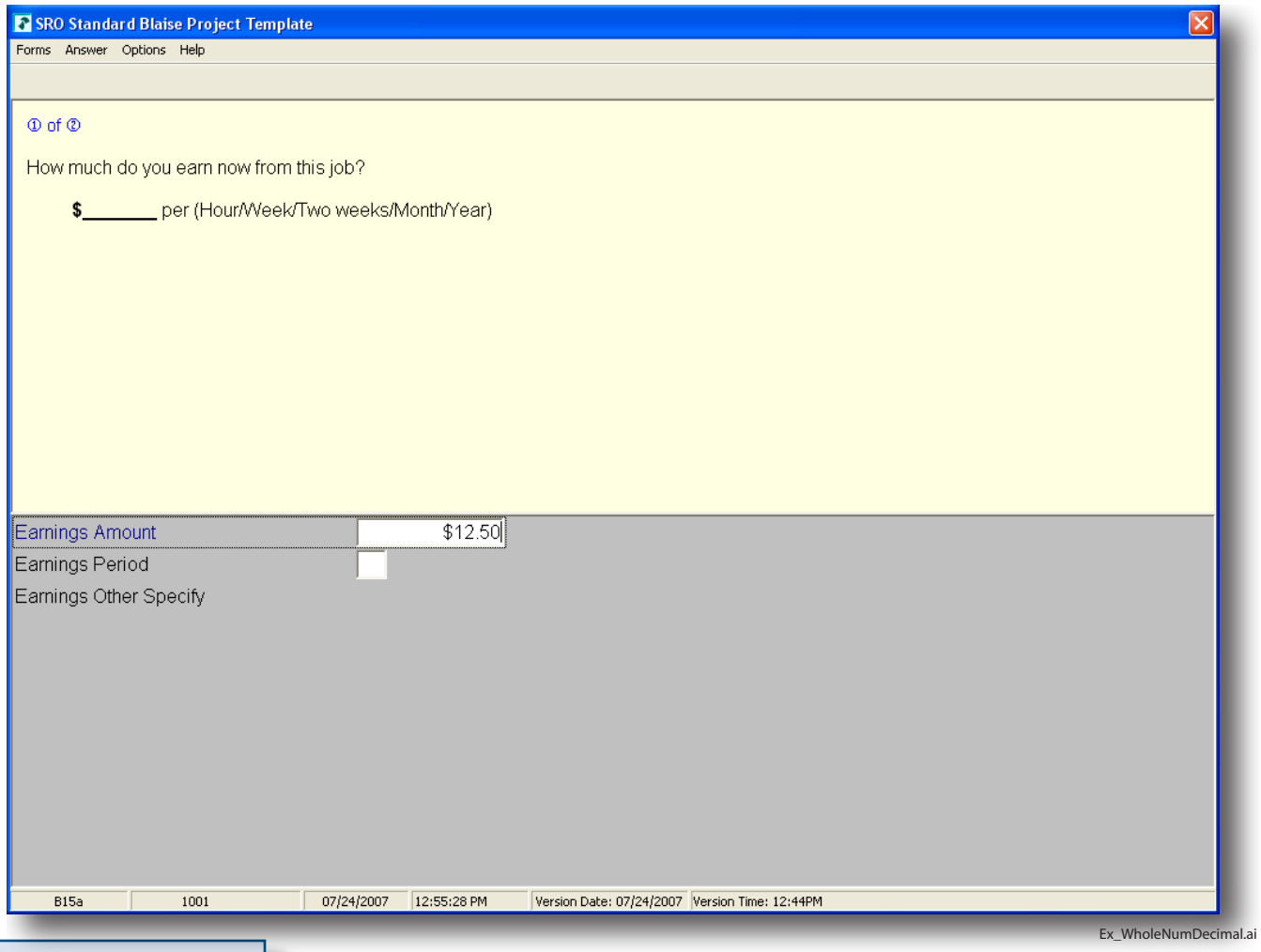


Ex_WholeNum.ai

Left-justified interviewer aid instruction precedes the question, in 11 point Arial blue, with 🕮 icon to identify it as a respondent booklet instruction.

Whole Number with Decimal

- Numeric response with one or more decimal places;



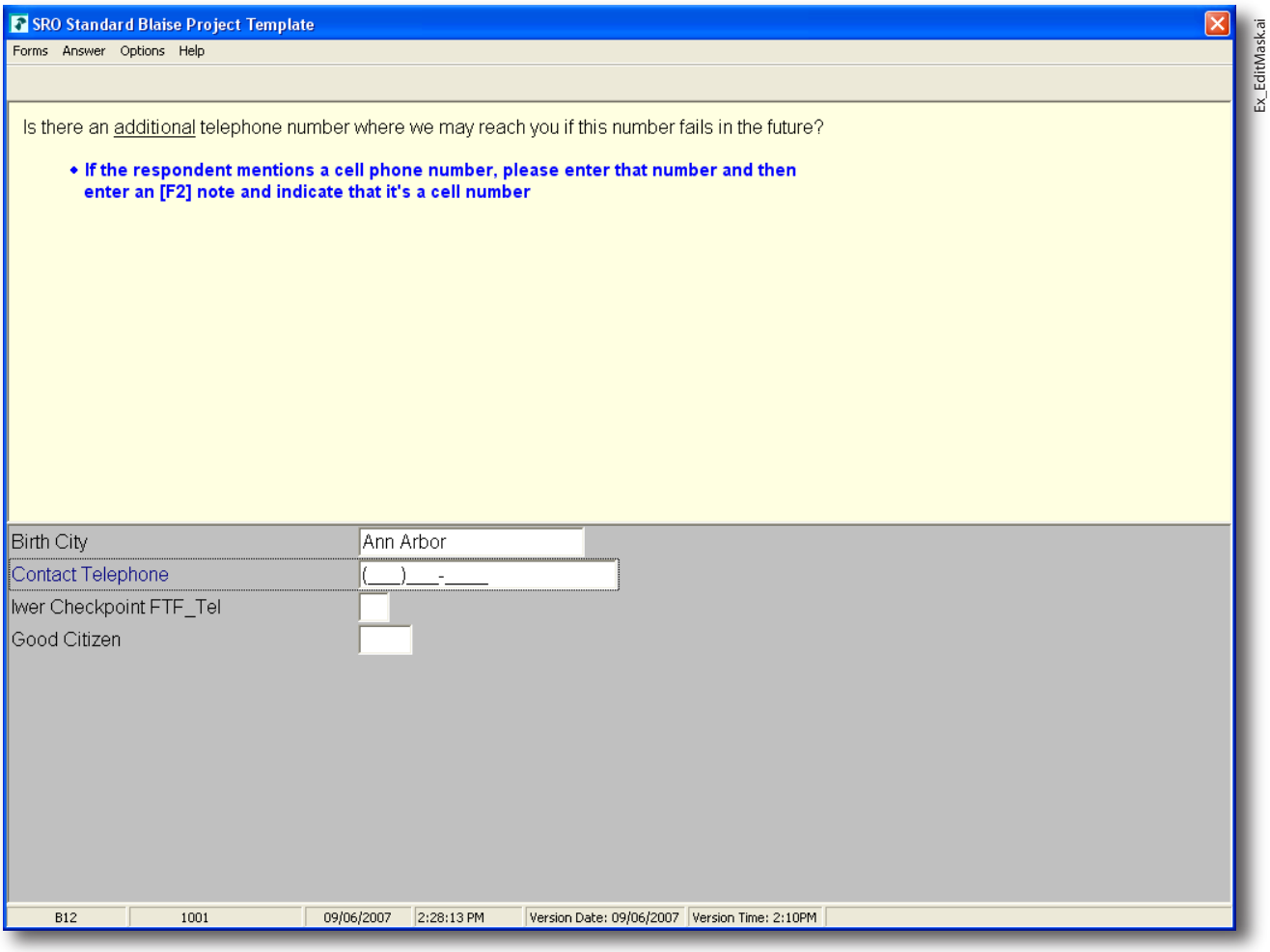
Ex_WholeNumDecimal.ai

"$" in Question and Entry Windows indicates to the interviewer that this is a currency response. The dollar sign in this entry window does not appear until the IWER starts to type

This example is the first of two questions in a multi-part question series (amount-per-period), and has a multi-part question indicator in the upper left corner.

Edit Mask

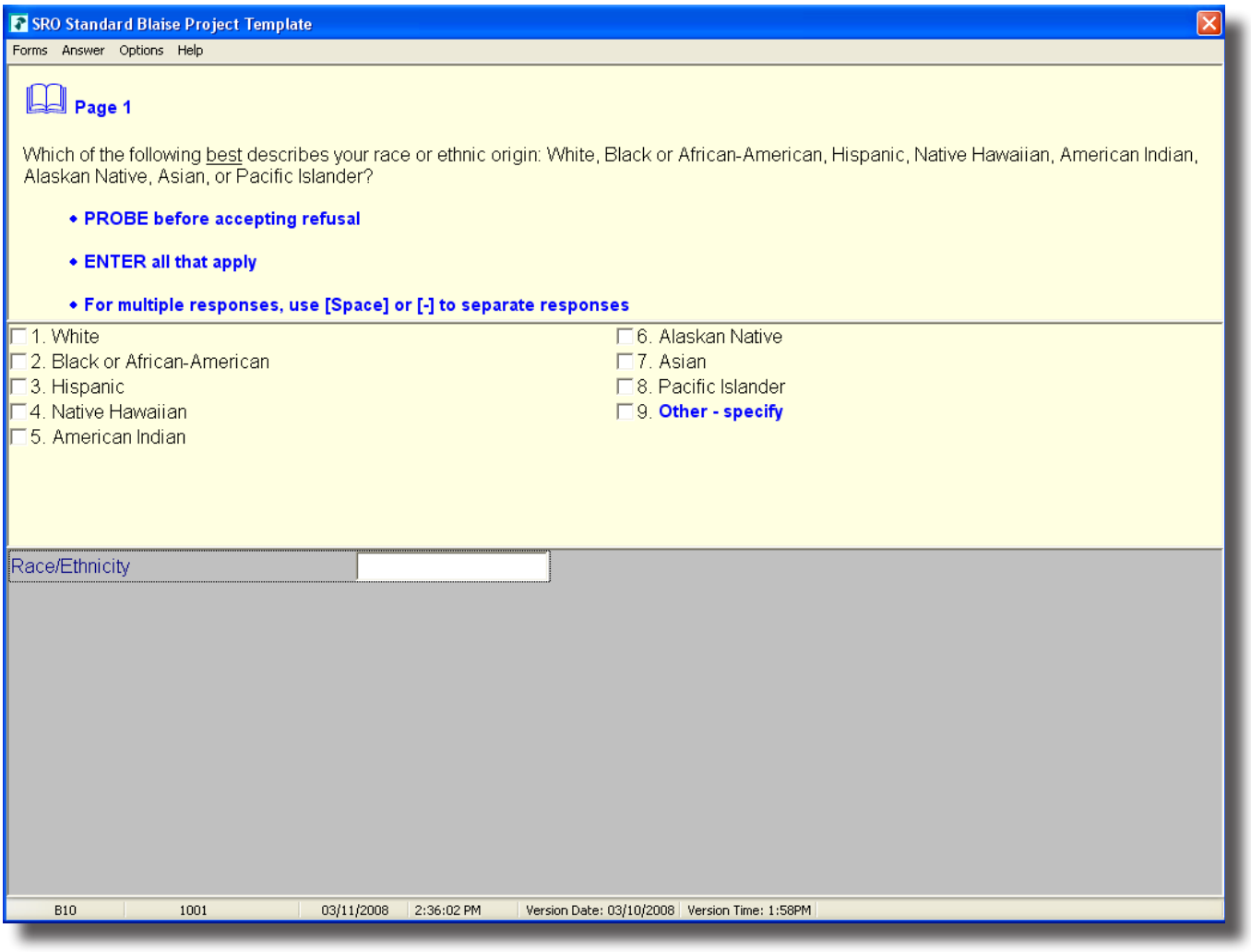
- Formatted field that provides guidance for data being entered, e.g., (___)___-____ for a telephone number.



Ex_EditMask.ai

Multiple Response

- Enter all responses that apply;

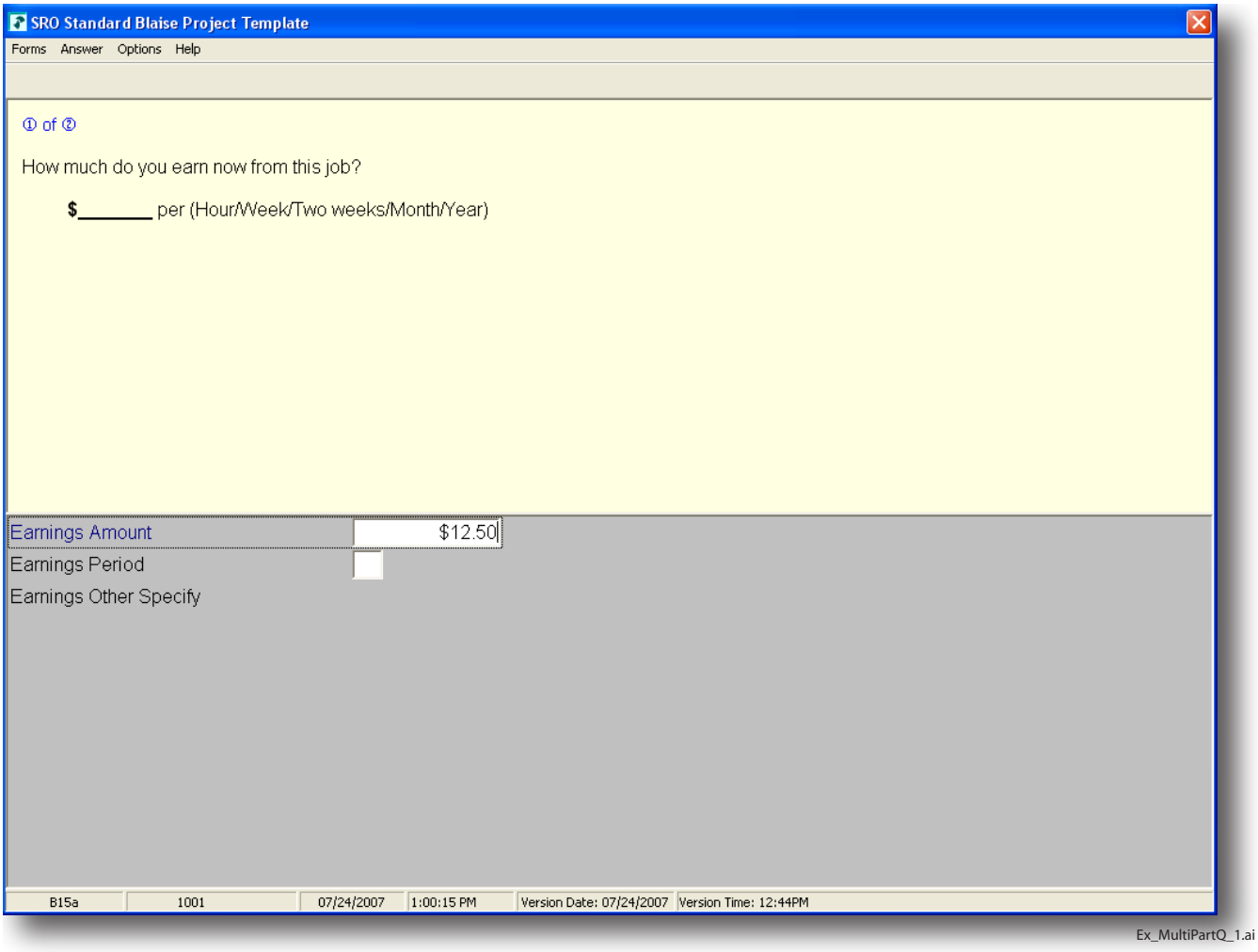


Note that instructions are double spaced, PROBE & ENTER are key action verbs and are capitalized, and that reference to key strokes are placed in brackets

In this example, the "Other - specify" category appears in blue, indicating it would not be read to the respondent.

Multi-part Questions (1 of 2)

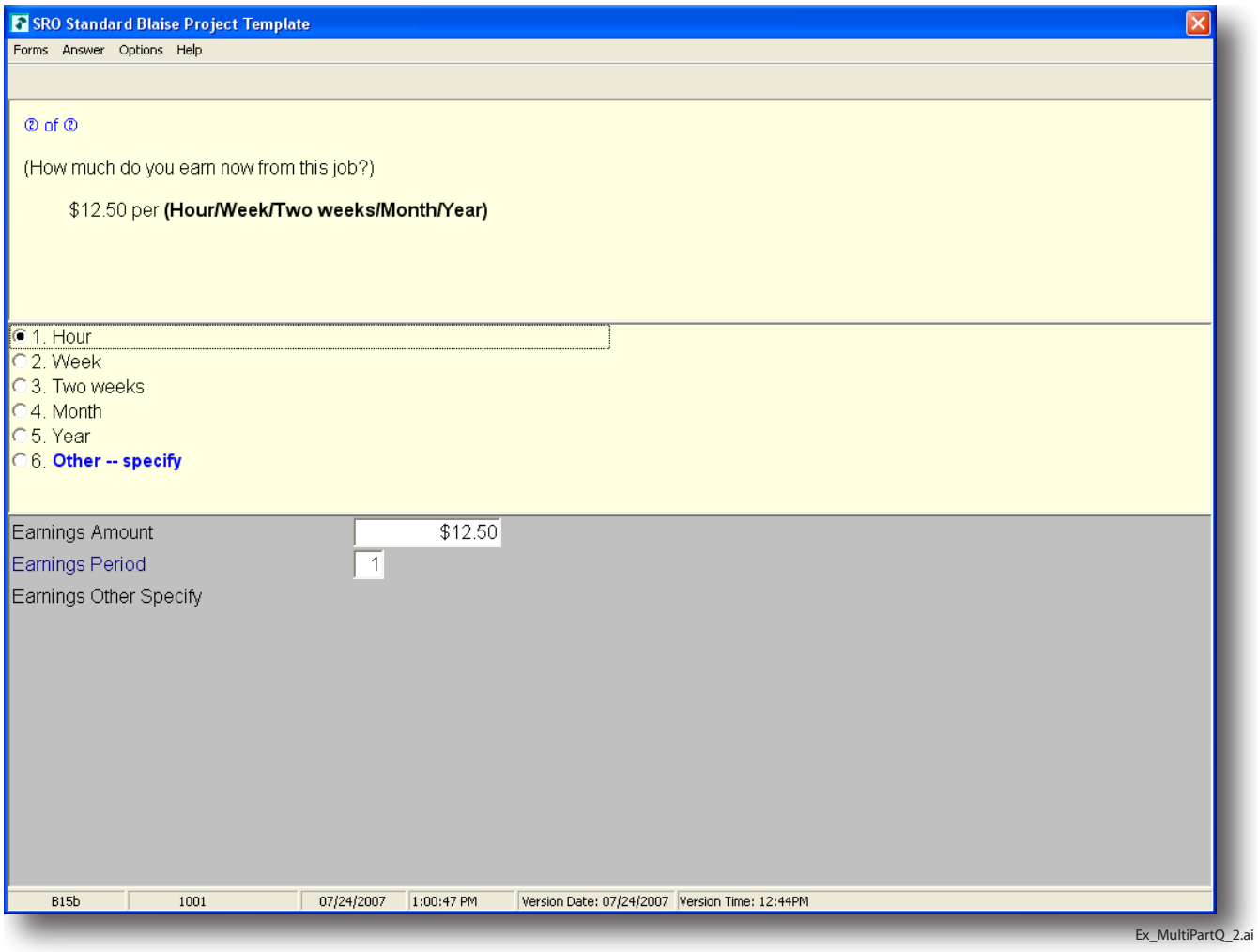
- The first screen of a multi-part question -- ① of ②;
- Screen task, e.g., enter dollar amount, is **11-point Arial bold blue**, and the remainder of the instruction is 11-point Arial blue.



Ex_MultiPartQ_1.ai

Multi-part Questions (2 of 2)

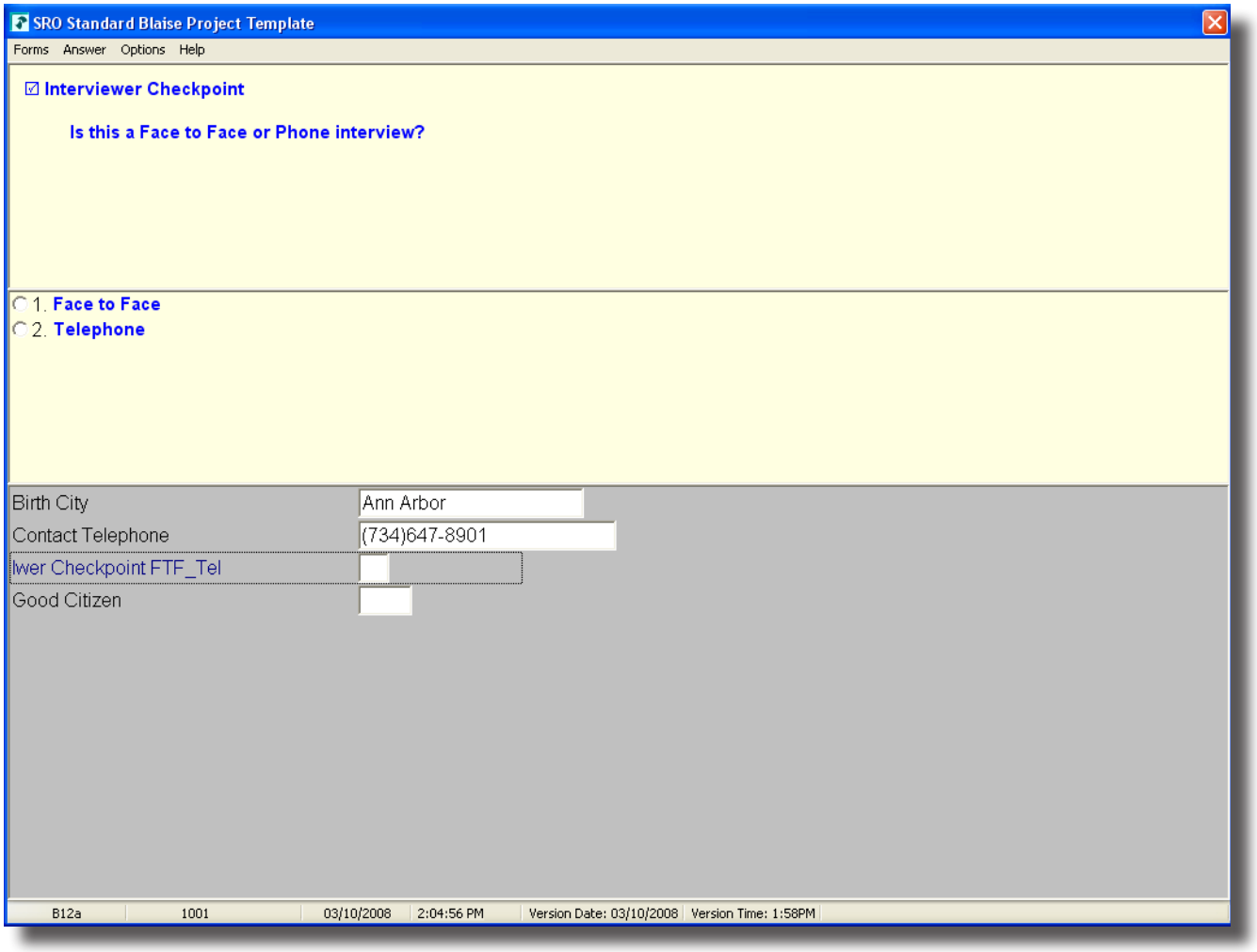
- The second screen of a multi-part question -- ② of ②;
- Screen task, (e.g., enter period), is **11- point Arial bold blue** ,and the remainder of the instruction is 11-point Arial blue;
- Response from the first part is filled in.



Ex_MultiPartQ_2.ai

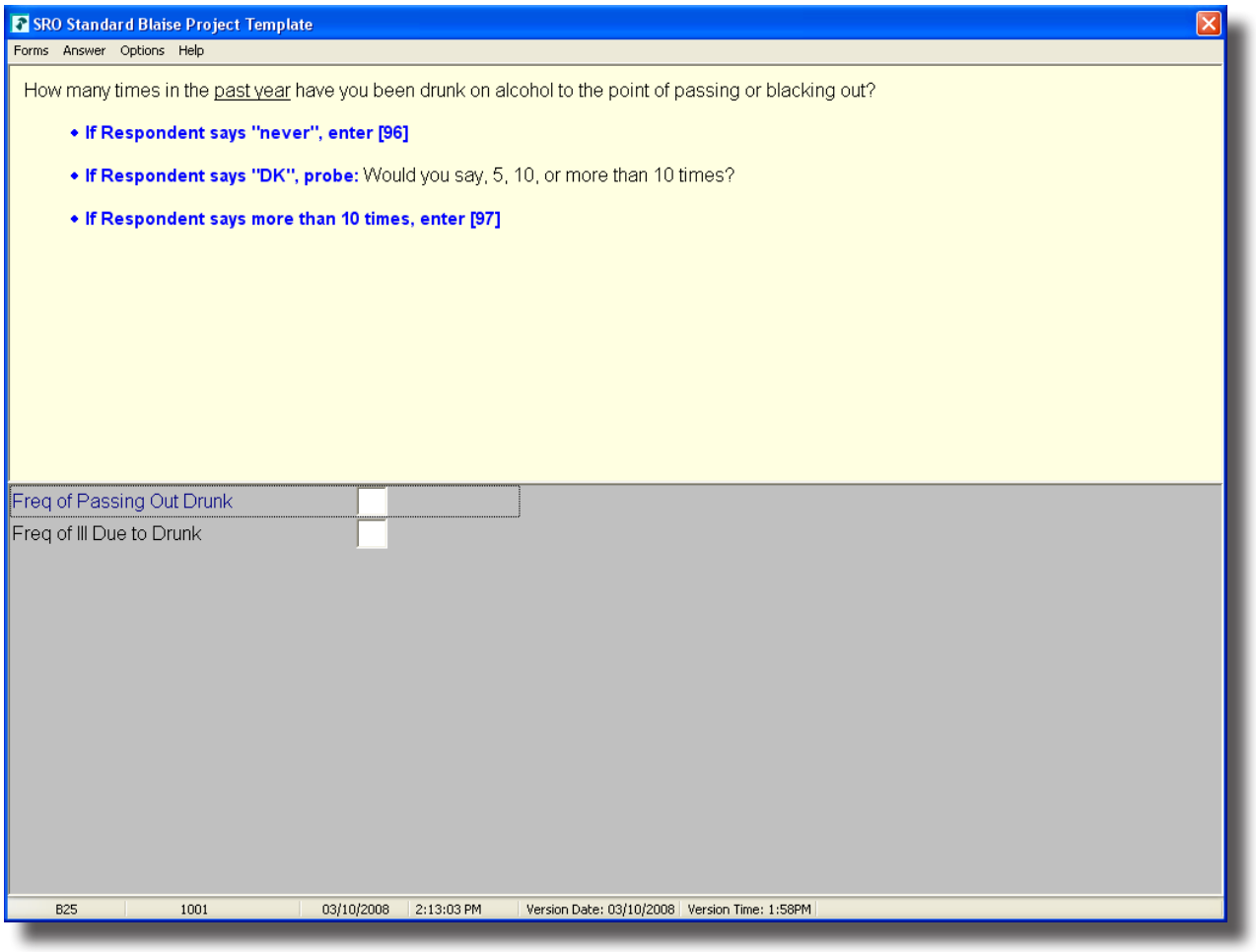
Interviewer Checkpoint

- An explicit interviewer checkpoint, with a question to the interviewer;
- All text of the interviewer checkpoint appears in **11-point Arial bold blue**;
- The special icon ☑ is used to distinguish an interviewer checkpoint.



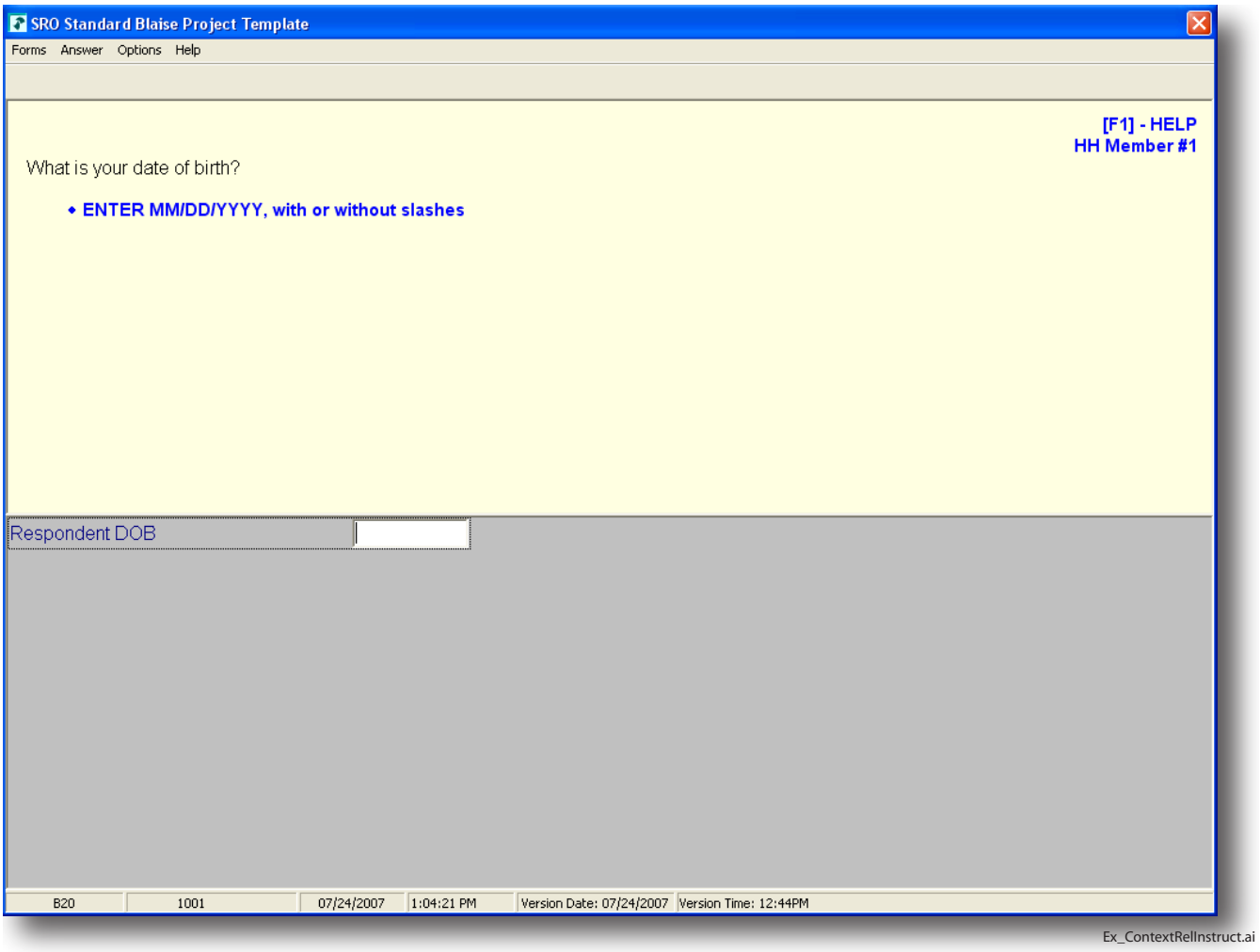
Special Interviewer Instructions

- In probe instructions, the text to be read is in 12-point Arial black;
- Reference to respondent answers appear in quotation marks;
- Conditional instructions start with a conditional phrase and action verbs (e.g., probe and enter) are not capitalized;
- Reference to numbers or keys to type are in square brackets.



Context-Related Instructions

- Context-related instructions are located on the right margin, below the question-level help indicator;
- Text should be in **11-point Arial bold blue**;
- Context-related items can be loop level indicators or any other information useful to the interviewer to give context for the question.

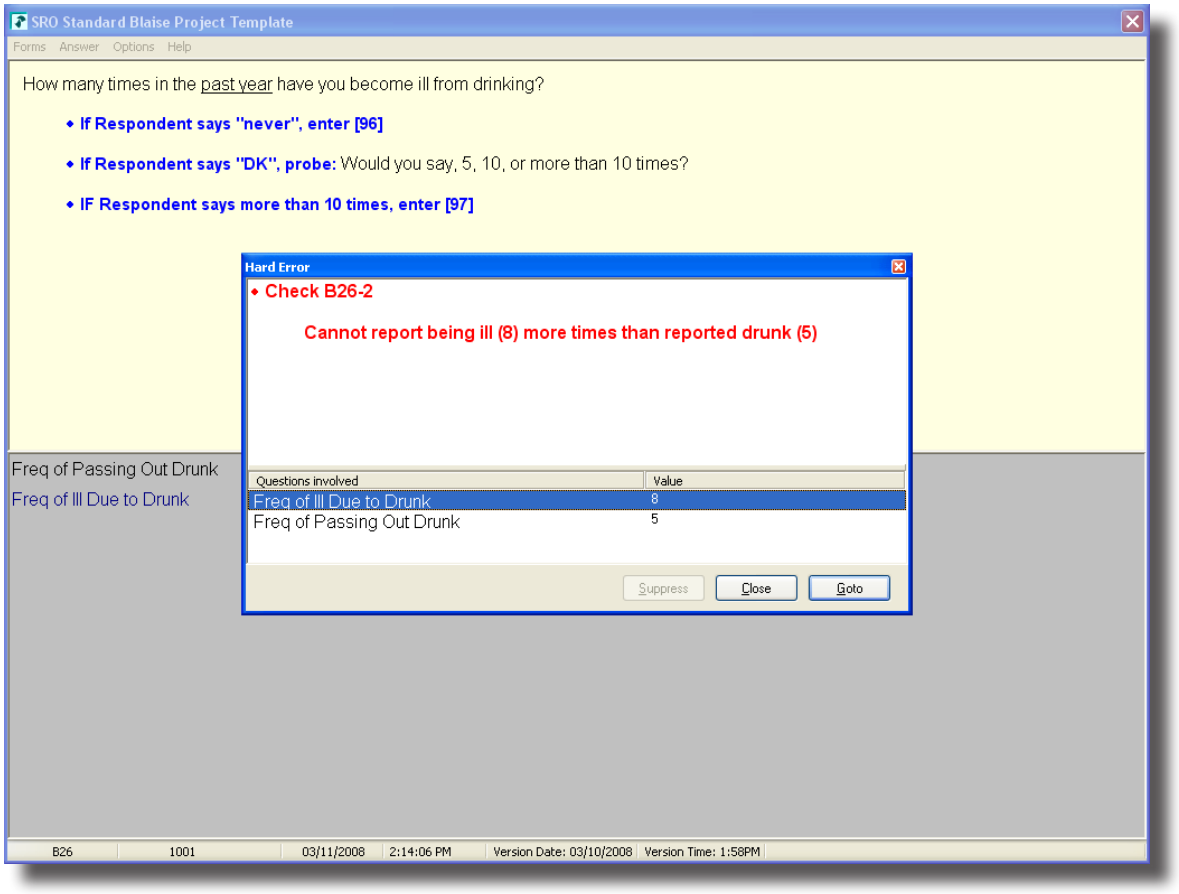


Ex_ContextRelInstruct.ai

This example shows a help indicator and context information in the upper right corner, and an indented data entry instruction following the question text.

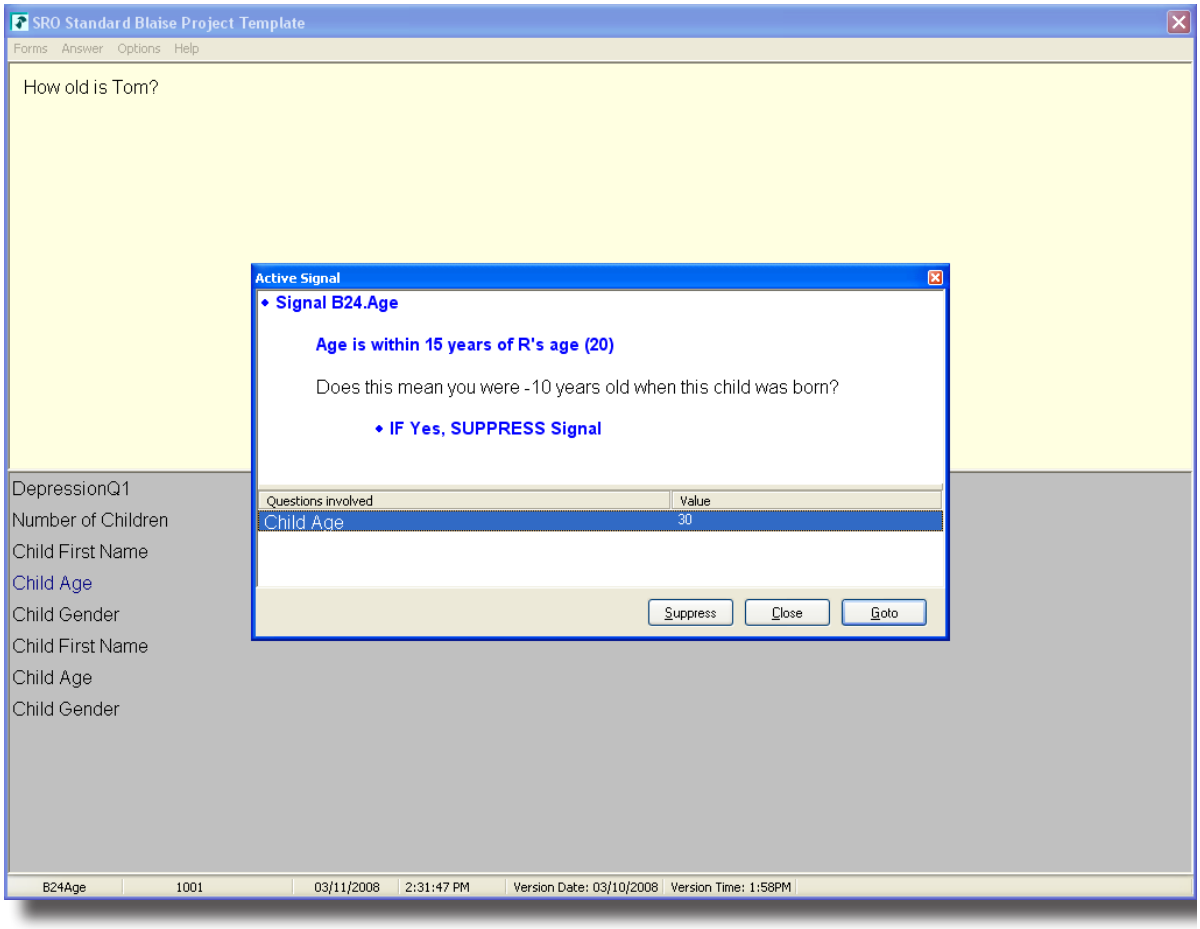
Hard Consistency Check (Check)

- Data consistency check that requires resolution;
- Triggers pop-up window with error message formatted as instruction, but in red, and prior question names and values evaluated as inconsistent with the current response;
- The interviewer either "goes to" highlighted question to re-ask it, or "closes" window to re-ask the current question;



Soft Consistency Check (Signal)

- Data consistency check that does not require resolution;
- Triggers pop-up window with error message formatted as interviewer instruction, and prior question names and values inconsistent with the current response;
- The interviewer either "goes to" highlighted question to re-ask it, "closes" window to re-ask the current question, or "suppresses" the consistency check to continue with the interview without changing any response;



This example shows a help indicator and context information in the upper right corner, and an indented data entry instruction following the question text.

Specification Standards

Introduction

CAI programming will reflect the study design requirements if those requirements are clearly specified. This chapter describes how to provide for the CAI programmer an overview of the entire instrument (through flowcharts and/or logical specifications), global instrument properties, and question-level specifications. The intent is to ensure that the instrument meets requirements of the study design, including data output and documentation requirements, and minimizes interviewer error.

Examples are provided of how to specify *common* types of questions found in survey instruments. It is not be possible to illustrate all question types. When you have a situation that is not explicitly covered here you should "take-your-best-shot" and follow up with a programmer to make sure that the specifications are complete and meet the study design requirements.

Specifications cover all aspects of design, including formatting for question text, interviewer instructions, and consistency checks. Thus, they reflect many of the SRC style preferences that are employed when developing a CAI questionnaire in Blaise, and as described in the Blaise Screen Design Standards chapter.

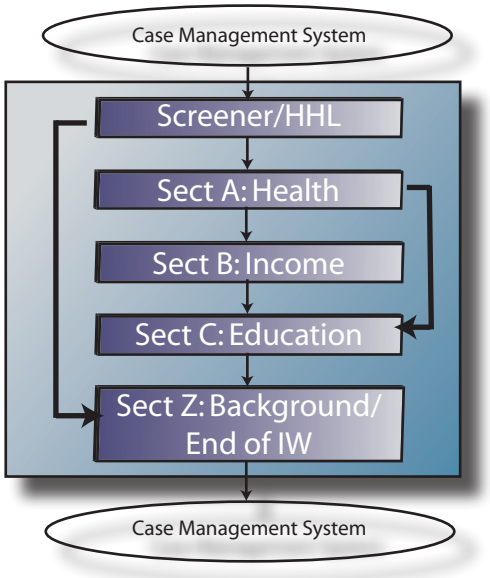
Specifications are used throughout the development and documentation processes. Having complete CAI instrument specifications not only helps programmers code more efficiently, but also allows testers to test more effectively. Specifications also serve as questionnaire documentation, and can be checked against the final Blaise instrument documentation and codebook generated by MQDS.

Creating Blaise Specifications

Instrument Overview

Specifications should start with an instrument overview. An overview provides programmers and project staff an overall sense of each CAI module used in the instrument. This may be in the form of a section level map (as shown blow) and/or a true flowchart. This is a useful because it helps to minimize misunderstanding about the overall flow of the instrument, and leads to a quick understanding about what instrument programming will entail. The display of questionnaire logic is not needed at this level; it is only intended to show overall instrument flow.

In addition to a section level flowchart or section map, the overview should include a list of additional features of the instrument that would impact the programming cost and timeline, such as number of languages, multimedia requirements, A-CASI components, event history calendar components, respondent selection algorithms, randomization, (e.g of sections, questions or response options) and recoding to be done within the Blaise instrument during the survey interview.



Specifying Global Project Attributes

Blaise allows setting global attributes that control missing data values (DK and RF) and whether questions may be skipped (EMPTY). For most instruments, these are specified as follows:

- Allow global Don't Knows (DK) and Refusals (RF) on all questions unless otherwise specified;
- Do not allow interviewers to skip questions (do not allow EMPTY) on any question unless otherwise specified.

Note that setting global DK and RF allows users to use the [CTRL-D] and [CTRL-R] hot keys on any question that does not have explicit missing data codes.

Instrument specifications should clearly state how these global attributes are to be handled (allowed or not allowed). In addition, specifications should include programmer instructions at questions where global settings should not apply, e.g.,:

- Unless otherwise specified, DK/RF follow the same routing as the "No" response
- Attributes: Do not allow DK/RF
- Attributes: Allow EMPTY

Default Characteristic

There are certain characteristics of a Blaise instrument that remain constant from project to project. These are *defaults* unless otherwise specified:

- DK/RF response will be treated the same as the negative/no response in the skip logic
- Dates will appear as: MM/DD/YYYY
- Telephone numbers will appear as: (###) ###-####

Preload

Preload are data that are loaded into the instrument prior to the execution of the interview. All Blaise instruments have standard preload, such as Sample ID and Interviewer ID. Each project may have additional study-specific preload requirements, such as:

- Address
- Telephone number
- Randomization or selection criteria data

It is important to identify the study-specific preload for the instrument before programming begins. When specifying preload fields, please provide the following information:

- Variable or field name (a maximum of 16 characters)
- Data type (e.g., Integer, String, Enumerated, Open End, or Date)
- Valid numeric range, if applicable

Project staff is responsible for creating preload. For pretesting, training, and data collection, project staff works with the Data Operations and Quality Control team member to setup and test preload.

Specifying A Blaise Question (1)

For each question in a Blaise instrument, generally the following are specified;

***Maximum Field Name character** length and **Maximum Field Description character** length are frequently overlooked. These lengths are important to note.*

1. Field Name (variable name), generally alpha-numeric (e.g. A1, B2, etc.). Note:
   - Avoid the use of underscores if possible, since field names are used in the Blaise program code and underscores increase programming time;
   - However, "ENTER all that apply" field names require underscores at the end (e.g., B1_);
   - Avoid the use of letters and numbers in positions where they may be confusing (such as the lowercase letter "l" and the number "1," letter "O" and number zero "0," the number "2" and the letter "Z,"; and
   - Due to variable naming constraints with some data processing software, ***Field Names should not exceed 16 characters***
2. Field Description, a brief meaningful descriptive text (e.g., Current Grade), ***not to exceed 25 characters in length***; this is what is displayed next to the entry window on the Blaise screen.
3. Question text
4. Response option categories (if an enumerated question)
5. Skip (go to) instructions
6. Data type (if not an enumerated question)

Note that a Blaise Field Tag will always be programmed; if not specified, it will be the same as the Field Name. As with Field Names, Field Tags have no spaces. Some surveys may need an additional variable descriptor or ID, and could use the Field Tag for this purpose. MQDS can display the Field Name, Field Description, or Field Tag as the variable name or ID, thus providing flexibility in generating documentation for a variety of purposes.

Particular questions may have additional specifications:

1. Online help indicator
2. Interviewer instructions, including probes
3. Optional text and variable text
4. Field-specific attributes (e.g., DK, RF, or EMPTY)
5. Edit mask
6. Logic for fills (variable text)
7. Explicit programmed checkpoint
8. Routing instruction logic
9. Logic for constructed or recoded variables
10. Soft consistency check (Blaise SIGNAL)
11. Hard consistency check (Blaise CHECK)

The next page indicates the order in which different question elements appear, and examples of specifications follows.

Specifying a Blaise Question (2)

NOTE: Standard screen design icons and instruction bullets and colors are not required in the specifications

Field Name {Alphanumeric, e.g., B1; no spaces}
Field Tag {= Field Name if not specified; no spaces}

n of n {e.g., 1 of 2; for multi-part questions, such as time-unit and period}

[F1] - Help\

RB Page #
Calendar
Interviewer Checkpoint

Interviewer instructions that precede question

Question text ^fill (optional text).?

Interviewer instruction(s)

Interviewer instructions that follow question

ENTER all that apply [if applicable]

/"Field Description" {Meaningful description with spaces}

Name1	Response option1 label	1	GO TO NextQ
Name2	Response option2 lsbel	2	GO TO NextQ
Name3	Response option3 label	3	GO TO NextQ
Name4	Response option4 label	4	GO TO NextQ
Name5	Response option5 label	5	GO TO NextQ
Other	Other - Specify	7	
DK			GO TO NextQ
RF			GO TO NextQ

Data type (mutually exclusive)

[Enumerated, (implicit); requires response options and relevant skips, as above]
Integer; range n-n; edit mask
Numeric; n decimal places, range n.nn-nn.nn; edit mask
Currency; n decimal places, range n.nn-nn.nn; edit mask
String; width= n; Edit Mask
Open End

Attributes: [DK, RF, EMPTY, NODK, NORF]

Routing instruction logic

Constructed variable or recode logic

Fill logic
Condition, "Fill text"
Condition, "Fill text"

Soft consistency check:
Condition
"Probe text"
Signal number; e.g. "Signal Fieldname"

Hard consistency check:
Condition,
"Probe text"
Check Number; e.g. "Check Fieldname"

Programmer Note
==========

Enumerated List, Multiple Response, Skip Logic, Online help (QxQ)

DA1_1

[F1]- -Help

Are you of Hispanic or Latino descent -- that is, Mexican, Mexican American, Chicano, Puerto Rican, Cuban, South or Central American or other Spanish culture or origin?

PROBE: Anything else?

ENTER all that apply

For multiple responses, use [Space] or [-] to separate responses

/"Hispanic or Latino Descent"

NotHisp	Not Spanish / Hispanic	1	GO TO D2
Mexican	Mexican	2	GO TO D2
MexAmer	Mexican American	3	GO TO D2
Chicano	Chicano	4	GO TO D2
PuertoRican	Puerto Rican	5	GO TO D2
Cuban	Cuban	6	GO TO D2
So_Central	South / Central American	7	GO TO D2
HispSpec	Other Spanish - Specify	8	
DK			GO TO D2
RF			

GO TO next question if "Other Spanish - Specify"

==========

Adding a visible break between questions helps programmers to clearly see what specifications are associated with particular questions

DA1_2
(Are you of Hispanic or Latino descent -- that is, Mexican, Mexican American, Chicano, Puerto Rican, Cuban, South or Central American or other Spanish culture or origin?)

ASK if necessary: Can you tell me what your Spanish ancestry is?

/"Hispanic or Latino Descent - Specify"
String (width = 40)

Enumerated List, Fills In Question Text

B22

Did you ever have a period of being [sad, discouraged or uninterested in things/sad or discouraged/sad or uninterested/sad/discouraged or uninterested/discouraged] that lasted most of the day, nearly every day, for two weeks or longer?

Yes	1	GO TO Sad30Days
No	5	
DK		
RF		

/"Sad Two Weeks or Longer"

IF B21a =1, THEN 'sad, discouraged, or uninterested in things'
IF B21b =5, DK or RF, THEN 'sad or discouraged'
IF B21c=1, THEN 'sad or uninterested'
IF B21d=5, DK or RF THEN 'sad'
IF B21e=1, THEN 'discouraged or uninterested'
IF B21f=5 DK or RF THEN 'discouraged'

==========

Respondent Booklet Instruction, Optional Text In Parentheses, Edit Mask

FN12

RB Page 55

The next questions are about the different sources of income you may have. For each question, please tell me the letter you see on page 55 in your booklet that represents the correct answer. First, what is your own personal earnings income in the past 12 months, before taxes? Count only wages and other stipends from your own employment, not pensions, investments, or other financial assistance or income. (Your best estimate is fine.)

ENTER $1,000,000 for answers greater than $1,000,000

/"Personal Earnings Past 12 Months"

Currency; 2 decimal places; Range 0.00-1,000,000.00; $n,nnn,nnn.nn

==========

Open End

EM10

Please tell me about the type of work you do.

PROBE: What kind of business or industry is that?

PROBE: What does this business make or do?

/"Industry"

Open End

==========

Date Type (Single Question Specified)

Multi-part dates are preferred since they allow DK/RF in separate fields

B20

What is your birth date?

ENTER MM/DD/YYYY, with or without slashes

/"Birth Date"

'1' on end of DateType indicates this question is one Blaise question, variable, and screen

DateType1; range for month 1-12; range for day 1-31; range for year1900-2007, MM/DD/YYYY

==========

Date Type (Multi-Part Question Specified)

Datetype3 data type requires Field Name of 14 characters or less

B20
(N of N)

Now I have some questions about your childhood. What is your date of birth?

ENTER date (MM/DD/YYYY)

ENTER [Ctrl-D] if DK day, month, or year

/"R's Date of Birth"

'3' on end of DateType indicates this date will appear as three questions

DateType3; Date Mask; Range for Month 1-12 Range for day 1-31 Range for year 1900-1984

Date Type (Multi-Part Question Programmed)

NOTE: While the previous shows **one question specified** for a multi-part question, DateType3 indicates to the programmer to create three questions and fill in prior responses , as shown below. Multi-part question indicators also will appear on the interviewer's screen. Specifications need only include components listed in the prior example (field name should be 14 characters or less to accommodate the suffixes of _M, _D, _Y)

B20Month
1 of 3

What is your date of birth?
MM/DD/YYYY
ENTER month

/"R DOB Month"
Integer; range 1-12

==========

B20Day
2 of 3

(What is your date of birth?)

08/**DD**/YYYY
ENTER day

/"R DOB Day"
Integer; range 1- 28/29/30/31, based on month and leap year status

==========

B20Year
3 of 3

(What is your date of birth?)

08/22/**YYYY**
ENTER year

/"R DOB Year"
Integer; range 1900-1984

==========

Arrayed Question Series (Index Question)

B23

How many natural born children do you have that are still living?

/"Number of Living Children"
Integer; range 1-20

Signal: (Signal B23)
IF CN1 > 10,
"Just be sure I am entering this correctly -- you have ^LivingChildren natural born children that are still alive; is this correct?"

Insert response to LivinChildren as fill in consistency check probe (^LivingChildren)

Arrayed Question Series (Looped Questions)

Display B24 series for number of children reported in CN1
==========

B24Name

What is the ^xFirstNext child's first name?

/"Child First Name"
String; width = 40

IF child = 1, THEN xFirstNext = "first", ELSE xFirstNext="next"

==========

B24Age

How old is [CHILDNAME]?

ENTER [1] if less than 1 year old.

/"Child Age"
Integer; range 1-20

CHILDNAME = Response at B24Name
==========

B24Sex

ASK if necessary: (Is [CHILDNAME] male or female?)

/"Child Sex"

Male	Male	1
Female	Female	2
DK		
RF		

==========

Checkpoints & Routing Instructions

Interviewer Checkpoint

SC9

Interviewer Checkpoint

Question flow based on response to question to interviewer (Interviewer Checkpoint); screen displayed but no question read to respondent

Is the respondent male or female?

/"Respondent's Sex"

Male	Male	1	GO TO SC31
Female	Female	2	

Attributes: NO DK, NO RF

Explicit Programmed Checkpoint

FD17
/"30-Day Functioning Criteria"

Explicit Programmed Checkpoint

Checkpoint based on programmed logic; explicit variable in data set (30Day_PCkpt); no screen displayed; no question asked of interviewer or respondent

(See FD2, FD2a, FD4, FD5, FD11a-d,FD13a-c,FD15a-d, FD17a-e)

Meets 30-day Criteria	30Day	1	

(FD2 Equals '1' – '2' AND FD2a Equals '3') OR
(FD4 Equals '3' – '30') OR (FD5 Equals '3' – '30') OR
(Two Or More Responses Coded '3' – '5' in
FD11a-d, FD13a-c, FD15a-d, FD17a-e)

All Others	Not30Day	5	GO TO FD23

Universe Statement for Routing

{ASKED IF AGE AT SCREENER <= 24 AND HAD SEX ED ABOUT "METHODS OF BIRTH CONTROL"}

BA_9
SEDBCG
What grade were you in when you first received instruction on methods of birth control?

ENTER [96] if R was not in school when he received the instruction

/"Grade Rec'd Inst on Birth Control Meth"

1st grade 1
2nd grade 2
3rd grade 3
4th grade 4
5th grade 5
6th grade 6
7th grade 7
8th grade 8
9th grade 9
10th grade 10
11th grade 11
12th grade 12
1st year of college 13
2nd year of college 14
3rd year of college 15
4th year of college 16
Not in school when received instruction 96

Routing Instructions

Routing instructions
(See AG1a-k)

Zero - One Responses Coded '1'	*GO TO AG39)*
Two - Three Responses Coded '1'	*GO TO AG3 INTRO 1*
Four Or More Responses Coded '1'	*GO TO AG3 INTRO 2*
All Others	*GO TO AG39*

Implicit programmed checkpoint; question flow based on complex logic, not just one enumerated response option; specification usually associated with a particular question, at the point it appears in the question flow; no data is stored, but MQDS will create an "IC variable," e.g., AG1_IC_12 as documentation of the routing instructions

Blaise Programming Standards

Program Start

At the very top of the program should be a comment block that has the following:

```
{************************************************************  }
{                          Study Name                          }
{                          Study Sub-Name                      }
{                          Main Application Control File (Bla) }
{PROGRAMMER:         Programmer Name                           }
{CREATED:            Creation Date                             }
{MODIFIED:           Modified Date                             }
{COPYRIGHT 2007 The Regents of The University of Michigan      }
{************************************************************  }
```

Programming Considerations - General

Use auxfields and locals as close in scope to where they are being used to avoid creating generated parameters. Certain elements, such as arrays, are unavoidable for generating parameters.

Programming Considerations - Blocks

In general, each instrument section or module should be programmed as a block by itself within an INC file. Subblocks are okay for arrays and such, but try to avoid too many nested blocks, if possible. Project staff may have a need to read the Blaise programs we create, and the clearer the programming code can be made, the better this will help us.

Preload should be put into its own separate block. The only preload variable that needs to be at the top level or the BLA file is SampleID.

If a section of code is used multiple times, or is in a loop, make it into a block. Put rules for determining the flow between sections at the uppermost level of the program (i.e., the rules in the BLA file). Put rules for determining flow in a section at the block level for that section.

Keywords

All keywords should appear in caps, i.e. DATAMODEL, IF, THEN, DO, TABLE, BLOCK, FIELDS, RULES. Do not use mixed/lower case for keywords, even if they are bolded in the program editor. Remember, when the program code is printed these words do not appear in bold.

Field Names

Field names should be as descriptive and meaningful as possible--for example, SpouseBirthDate instead of Bdat. Each word of the description is capitalized. These conventions generally apply to the naming of Blaise fields or variables:

1. Underscores have to be used on Fields that are SET (Multiple Choice) fields; e.g. MedsTaken_. Avoid them in the middle of names (e.g., Meds_Taken), since they make programming Blaise Rules more difficult.
2. Keep them to 16 characters or less, to minimize amount of typing of Blaise code, and to accommodate variable name constraints in data processing software.
3. Avoid the use of letters and numbers in positions where they may be confusing (such as the lowercase letter "l" and the number "1," letter "O" and number zero "0," the number "2" and the letter "Z," etc.

Field Descriptions

Place field descriptions on their own line and in Title Case. For example, the "R's Date of Birth" is the field description. Field Descriptions should be meaningful yet restricted to 25 characters or less because they are used in the Field Pane instead of the Field Name.

```
RbirthDate          (A3)
                    "What is your date of birth?" /
                    "R's Date of Birth" :

                    TDate
```

Be careful of using field descriptions on blocks – each occurrence will appear on the screen in the upper left hand corner of the Info pane when used on the block itself. It is not recommended to do the following:

```
MySection           / "Section MySection" :
                    BMySection
```

However, they are desirable for documentation. For this purpose, we recommend that a separate datamodel with block descriptions be maintained.

Hungarian Notation

The following are the naming conventions to be used (the word in Bold is how the typename should be used):

General Type:	TYesNo = (Yes, No)
	TDate = DATETYPE
Tables:	TABLE THousehold
Blocks:	BLOCK BPerson
Sets:	MySet_ : SET OF MyType
Auxfield:	xTempName : STRING[20]
Field:	AnyField : TNameType
Input Parameter:	piName : STRING
Export Parameter:	peName : STRING
Transit Parameter:	ptName : STRING

Type Use

Types in a separate project-specific type INC file should always be used to ensure consistency throughout the program.

Remember, for this situation

```
MyYesNo              : TYesNo
YourYesNo            :(Yes (1),
                       No  (5))
```

you will get a type mismatch error (on enumerateds) if you check MyYesNo = YourYesNo. This error does not occur for ranges.

Types also allow input masks to be placed on them, making them even more useful.

Code Names

The names of the enumerated list categories should be mixed case, and followed by descriptive text when more than 1 word is necessary for the Code Label. For example,

```
TNumBooks = (None,
             OneTwo "One or two",
             ThreetoNine "3 to 9",
             Tento19 "10 to 19",
             TwentyPlus "20 or more")
```

Multiple Languages

When programming multiple languages, put each new language on its own line, lining up with the text on the prior line. For example,

```
A34     (A34)
        "@>@I[F1] - HELP@I@<
        @/How is that?

        @/@/(Could you tell me a little more about your situation?)"

        "@>@I[F1] - HELP@I@<
        @/¿Cómo es eso?

        @/@/(¿Podría Ud. decirme un poco más acerca de su
        situación?)"/

        "No Own No Rent-Specify" :

        STRING[400]
```

When more than 2 languages are used, then each language should be labeled appropriately in the code. For example:

```
A34             (A34)
        ENG     "@>@I[F1] - HELP@I
                @/How is that?

                @/@/(Could you tell me a little more about your
                situation?)"

        SPAN    "@>@I[F1] - HELP@I
                @/¿Cómo es eso?

                @/@/(¿Podría Ud. decirme un poco más acerca de
                su situación?)"

        GER     "@>@I[F1] - HELP@I
                @/Wie ist das?

                @/@/(Konnten Sie wenig mehr erklären mir über
                ihre Situation?)" /

        "No Own No Rent Specify"

        STRING[400]
```

Global DK/RF

Global DK/RF is set for every question. This is accomplished by putting the following line of code in the BLA file after the "DATAMODEL" statement:

```
DATAMODEL FirstNew "First questionnaire"

ATTRIBUTES = DK, RF
```

Discontinuous Ranges

If a question requires multiple ranges (i.e., ages 1..120, 998..999), then the range should be put in full (1..999) and a hard check be used to exclude invalid values.

Calculations/Constants

All global constants should be declared at the very start of the program to make changes easy to do. Modular ("include" files) constants should be at the very start of module.

For example, get rid of the constant (1980, 1, 1)

```
IF (BirthDate > (1980,1,1)) THEN
        CHECK
                Married = No "^Name is too young to be married"
ENDIF
```

and instead declare it early in the program by using a calculation:

```
cAge18YearsAgo: 0..120
...
cAge18YearsAgo := STARTDATE - (YEAR(STARTDATE)
-18,MONTH(STARTDATE),DAY(STARTDATE))
```

Locals

Variables such as reused arrayed indexes should be stored as locals.

```
LOCALS
        I   : INTEGER      {Arrayed index if necessary}
```

Things to Always Store in the Data

```
AUXFIELDS
    {Auxfields Needed for Back-Up Timing Code}
    xTime           : TIMETYPE
    xDate           : DATETYPE

FIELDS
    {PRELOADED VARIABLES BEGIN}
    SampleID        (SampleID) / "Preload-Sample ID" : TSampID

    {Preload Block contains all preload vars for project}
    Preload         (Preload) / "Preload Block" : BPreload
    {PRELOADED VARIABLES END}

    VersionDate     (VersionDate) / "Application Version Date" : DATETYPE

    VersionTime     (VersionTime) / "Application Version Time" : TIMETYPE

    {Fields Needed for Back-Up Timing Code}
    CurrentSecs     (CurrentSecs) / "Current Seconds"   : INTEGER
    TotalSecs       (TotalSecs) / "Total Seconds"       : INTEGER
    TotalMins       (TotalMins) / "Total Minutes"       : 0.00..60000.00

    IWStartDate     (IWStartDate) / "Interview Start Date"  : DATETYPE

    IWCurrDate      (IWCurrDate) / "Interview Current Date" : DATETYPE

    TS_BeginIW      (TS_BeginIW) / "Interview Start Time"   : TIMETYPE

    Section_A       : BSection_A

    Section_B       : BSection_B

    Section_C       : BSection_C
```

IW Sections are inserted here

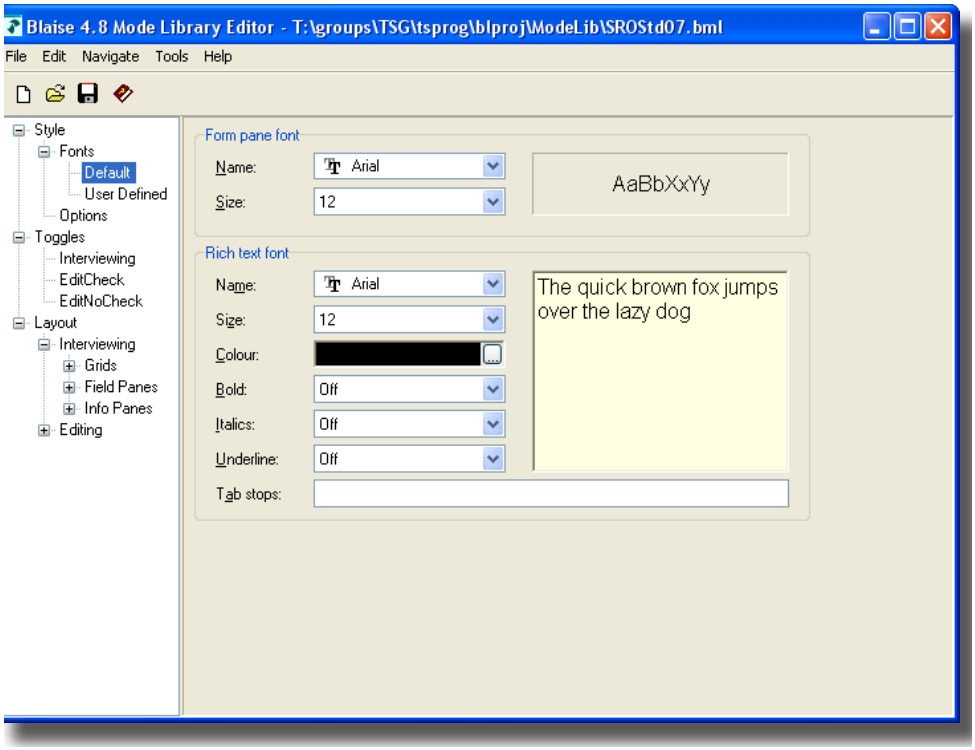
```
    IWComplete      (IWComplete)
                    "@/@Dw@D @IYou have reached the end of the Interview@I

                    @/@/@|@Dw@D @IENTER [1] to complete@I" /
                    "Interview Complete" :

                    TComplete, NODK, NORF

    Complete        (Complete) / "Complete Var for ST/SMS" : (Done, NotDone)

    TS_EndIW        (TS_EndIW) / "Interview End Time" : TIMETYPE

    IWEndDate       (IWEndDate) / "Interview End Date" : DATETYPE

    IWTotal         (IWTotal) / "Time Total Of Interview" : INTEGER

RULES
    SampleID.KEEP
    Preload.KEEP

    VersionDate.KEEP
    VersionDate := TODATE (2007, 04, 20)

    VersionTime.KEEP
    VersionTime := TOTIME (15, 44, 00)

    {Back-Up Timing Vars}
    xTime.KEEP
    xDate.KEEP
    CurrentSecs.KEEP
    TotalSecs.KEEP
    TotalMins.KEEP

    IF xTime = EMPTY AND xDate = EMPTY THEN
        xTime := SYSTIME
        xDate := SYSDATE
        TotalSecs := TotalSecs + CurrentSecs
        CurrentSecs := 0
    ENDIF

    CurrentSecs := (ABSTIME(SYSTIME) - ABSTIME(xTime) + 86400000 * (SYSDATE - xDate)) / 1000

    {INTERVIEW START DATE}
    IWStartDate.KEEP
    IF IWStartDate = EMPTY THEN
        IWStartDate := SYSDATE
    ENDIF

    {INTERVIEW CURRENT DATE}
    IWCurrDate := SYSDATE

    {INTERVIEW START TIME}
    TS_BeginIW.KEEP
    IF TS_BeginIW = EMPTY THEN
        TS_BeginIW := SYSTIME
    ENDIF
```

```
      INTERVIEW SECTIONS HERE

    IWComplete

    IF IWComplete <> EMPTY THEN
        Complete := DONE
    ENDIF

    {Back-Up Timing Data if ADT timings fail;
     this timing sequence works regardless of suspends}
    IF Complete = DONE THEN
        TotalSecs := TotalSecs + CurrentSecs
        TotalMins := TotalSecs / 60
    ENDIF

    {INTERVIEW END TIME}
    TS_EndIW.KEEP
    IF IWComplete <> EMPTY AND TS_EndIW = EMPTY THEN
        TS_EndIW := SYSTIME
    ENDIF

    {INTERVIEW END DATE}
    IF IWComplete <> EMPTY AND IWEndDate = EMPTY THEN
        IWEndDate := SYSDATE
    ENDIF

    {INTERVIEW LEGNTH CALCULATION}
    {Overall Interview Legnth In Seconds Calculation}
    {This calculation only good if IW is completed without
any suspends}
    IWTotal.KEEP
    IF TS_BeginIW <> EMPTY AND TS_EndIW <> EMPTY THEN
        IWTotal := (ABSTIME(TS_EndIW) - ABSTIME(TS_
BeginIW))/1000
    ENDIF

    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
    RESERVECHECK
```

Ten (10) Reserve Checks should be put into the BLA and each successive INC file as placeholders for future use if necessary.

Question Text Size

In the Mode Lib Editor, and the DEP Config file (DIW), Under Style – Fonts – Default, you can set the font type and size. The SRO standard type and size is Arial in 12 point.



Balancing Response Categories

When using more than seven (7) response categories, it is advisable to use multiple columns and balance them out into even or close to even columns. In the configuration editor choose Layout – Interviewing – Info Panes – InfoPaneName (this is the name of the custom Info Pane you are using with response categories that require multiple columns). In the Controls section of the "Layout" tab, you will see three areas: Field Text, Answer List, and Answer Info. Under Answer List, there is a check box for "Balance Answers". Make sure that box is checked.

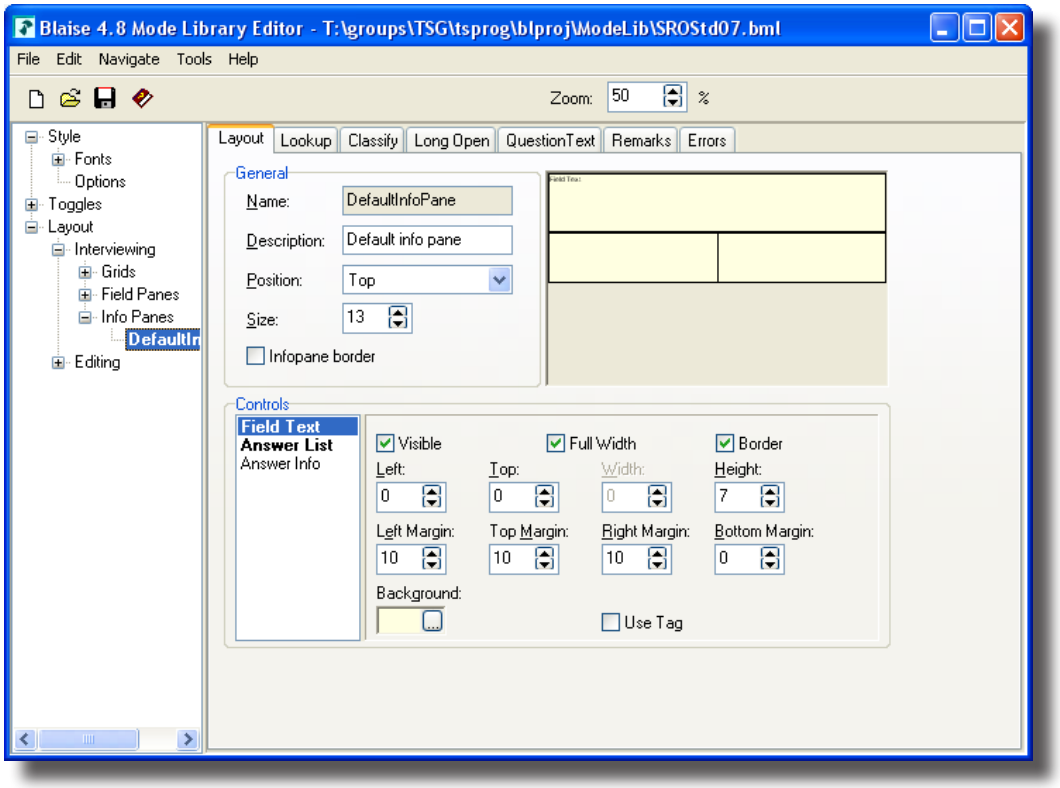
Note: There is a standard SROStdModelib.bml you can find it here:

\\Src-rensis\vol1\blaise\ModeLib

All projects should use this mode library as a start for their project, and copy it to their project directory & modify it to fit their needs or just make adjustments via the DEP configuration file (.DIW).

Info Pane Margins for Question Text

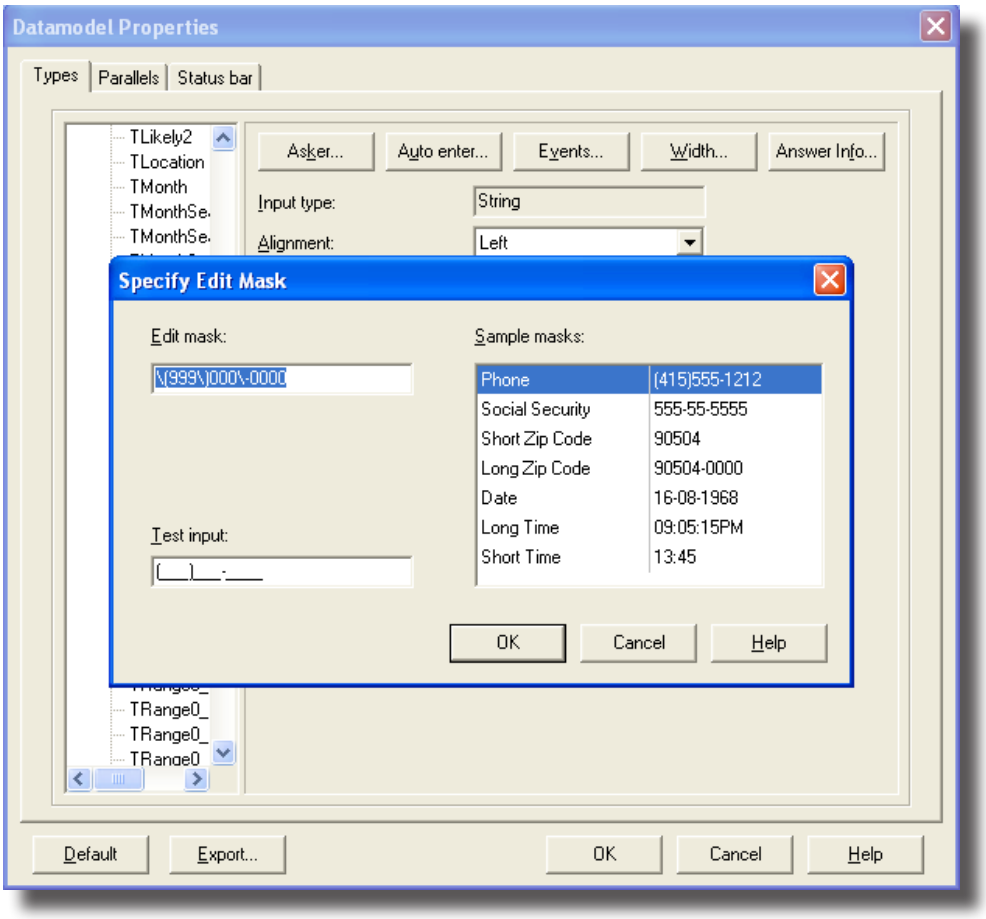
Info Pane Margins should be set in the Mode Library to produce the following: Top Margin = 10 Pixels from top, Left Margin = 10 Pixels from left edge, and Right Margin = 10 Pixels from right edge. Example of how to set is below under the Layout – Interviewing – Info Panes – InfoPaneName (this is the name of the custom Info Pane you are using with response categories that require multiple columns). In the Controls section of the "Layout" tab, you will see three areas: Field Text, Answer List, and Answer Info. Under Field Text, there are three boxes that control the Top Margin, Left Margin, and the Right Margin. Set all of these at ten (10).



Edit Masks

Edit Masks, although very useful, can only be invoked on String Fields that has been defined as a *Type*.

There are a variety of sample edit masks available and you can create your own too. To set an Edit Mask: in the Control Center, choose Project – Data-model Properties. In the types, select the User Defined type you want to specify an Edit Mask (for this example, we are using TPhone). At the Edit Mask definition line, click on the Specify button. This will take you to a dialog box where you are able to specify the type of Edit Mask you want to use. There is an Edit Mask available for Phone. Highlight it and click on OK.



Program Format - Comments

ENDBLOCK, ENDCASE, ENDCLASSIFICATION, ENDPROCEDURE, ... always should have a comment following describing what is ended.

```
BLOCK BPerson
.
.
.
ENDBLOCK {BPerson}
```

Describe IF statements and FOR loops, use the ending comments as necessary, especially at the end of long sections of conditions.

```
IF MyAge < MyBirthDate THEN
.
.
.
ENDIF {MyAge < MyBirthDate}
```

Comments should precede complex pieces of code, blocks (with or without parameters), or other like situations on the preceding line(s). They should be indented at the same level as the program code being used.

Program Format - Field Text

Do not use @/ for every line because we set the margins in the modelib; use them only to separate paragraphs. Indent lines as needed by using the Tab ("@|"); if needed, you can define tab stops in the Fonts section of the modelib. Use CTRL-period-period to put in two hard spaces after the period at the end of the sentence – using two normal spaces will just lead to one space displayed on the Blaise info pane.

Break each paragraph of information out in the editor so that it's easier to read the program code, and to have an inkling of what the display of the question may be like.

Program Format - Display width

For ease of reading your program when printed, keep the program code within the "80 column" display width of the editor. You can see what column you're in by looking at the lower left of the editor pane; it shows the row and column number of the cursor. Using these guidelines below for spacing your code enhances its readability significantly; especially for those readers who do not deal with Blaise code on a daily basis.

- Declare your "BLOCK," "FIELDS," "RULES," "LAYOUT," and "ENDBLOCK" keywords flush with the left side of the editor
- Tab in 1 tab stop to column 5 to place your Field Name
- Tab over to column 21 to place the Field Tag, Field Text, Field Description, and Field Type
- Place the slash (/) to separate the Field Text and the Field Description one space after the double-quotes that end the Field Text.
- Place the colon (:) that comes between the Field Description and the Field Type at the end of the Field Description and double-space prior to declaring the Field Type.

Example:

```
FIELDS
       J1     (J1)
              "How long have you lived in your current
neighborhood?" /
              "Length of Residence 07" :

              TDuration
```

Program Format - Line Spacing

Put an extra blank line between groups of related statements. For example, do not do

```
BLOCK BFirst
ENDBLOCK {BFirst}
BLOCK BSecond
ENDBLOCK {BSecond}
```

But use instead

```
{This is about the first Block}
BLOCK BFirst
<blockstuff>
ENDBLOCK {BFirst}

{This is about the second Block}
BLOCK BSecond
<blockstuff>
ENDBLOCK {BSecond}
```

and ...

```
{Comment about compound IF stuff below}
IF condition1 THEN
    IF condition2 THEN
        <dostuff>
    ENDIF {condition2}

    {Comment about this IF statement}
    IF condition3 THEN
        <dostuff>
    ENDIF {condition3}
ENDIF {condition1}
```

In general, if there's an ENDIF statement that is followed by something other than another ENDIF, put a blank line after the former. Using more blank lines is better than using too few.

Program Format - Indenting

Keywords DATAMODEL, ATTRIBUTES, FIELDS, RULES, ... should start in the leftmost column.

```
DATAMODEL
USES
EXTERNALS
ATTRIBUTES
LOCALS
TYPE
AUXFIELDS
FIELDS
RULES
ENDMODEL
```

TABLE, BLOCK, ENDTABLE, ENDBLOCK, PROCEDURE, ENDPROCEDURE ... should appear one tab stop in (column 5) from the left margin. Each additional level of the program should be indented one more tab. A tab of 4 spaces is used in the example below.

```
DATAMODEL MyModel "Description of MyModel"

FIELDS

    BLOCK BPerson
```

```
        FIELDS
            FieldName1   (Field 1 Field Tag) /
                         "Field 1 Field Description" :

                         TType

            FieldName2   (Field 2 Field Tag)
                         "Field 2 Field Text"

                         "Field 2 Alternate Field Text
                           (e.g. Spanish)" /
                         "Field 2 Field Description"
                         "Field 2 Alternate Field Description" :

                          TField2

        RULES
            FieldName1
            IF FieldName1 = RESPONSE THEN
                FieldName2
            ENDIF    {FieldName1 = RESPONSE}

    ENDBLOCK    {BPerson}

    TABLE THousehold
        LOCALS
            I         : INTEGER
        FIELDS
            Person    : ARRAY[1..10] OF BPerson
        RULES
            Person[1]
            FOR I := 2 TO 10 DO
                IF Person[I - 1].FieldName2 = YES THEN
                    Person[I]
                ENDIF
            ENDDO
    ENDTABLE    {BHousehold}
FIELDS
    PersonInfo      : THousehold

RULES
    PersonInfo

LAYOUT
    BEFORE PersonInfo NEWPAGE

ENDMODEL {MyModel}
```

Program Format - Field Names

```
FIELDS
    TownName        (A12)
                    "What town do you live in?" /
                    "Town of Residence" :

                    STRING[20]

    Fears           (A13)
                    "What scares ^xName most?" /
                    "Most Scary" :

                    TScare
```

Program Format - Field Descriptions

Field Descriptions should always be placed on a line by themselves.

Program Format - Arrays and Indices

Do not place spaces between a field name and the index, or any part of the index. For example, always use this format:

Use

```
MyFieldName[1..10]
```

Do NOT use (any variation using extra spaces)

```
MyFieldName [ 1..10 ]
```

Program Format - Parameters & Attributes

Place an extra space after the use of a comma

```
FIELDS
    MyField          (MyFieldTag)
                     "This is MyField" /
                     "MyField Field Description" :

                     0..21, EMPTY, NODK, REFUSAL

RULES

    RunMyProcedure(ParamOne, Param2, Param3)
```

Program Format - Tags

Tags follow the field name within parentheses. Most likely these tag names will be the short description labels.

```
A12                  (TownName)
                     "What town do you live in?" /
                     "Town of Residence" :

                     STRING[20]
```

Program Format - Fills

The general standards for fills are:

Fills are unique per instance used and unique per language. Also, fills should be declared as close to where they are to be used as possible. This helps in program efficiency and reduces the nuimber of generated parameters in the program. It also lends heavily to producing cleaner documentation.

For example, use fills xhe_she, xel_ella, xis_are, xes_son instead of just one generic fill.

```
IF A1 = Male THEN
       xhe_she := 'he'
       xel_ella :='ella
ELSE
```

```
            xhe_she := 'she'
            xel_ella := 'ella'
      ENDIF
      A2

      IF A2 = 1 THEN
            xis_are := 'is'
            xes_son := 'es'
      ELSE
            xis_are := 'are'
            xes_son := 'son'
      ENDIF
A3
```

Two Major Issues to Avoid When Constructing Fills

1. Re-assigning Fills [DO NOT USE UNDER ANY CIRCUMSTANCE]

Re-assigning fills (scope of a fill) presents a particular problem for displaying correctness in MWDS. Because assignments can be made to a fill at any time before it is used, and because the assignments can totally replace all conditions for the prior fill, it will appear correctly to the interviewer but is very hard to determine by a question which assignments should be displayed.

2. Fills in Arrays

Fills in arrays are absolutely necessary for sorting members within households, restricting lists of items (on enumerated lists), etc.

```
xFill[i] where i can vary from 1 through 20.

This, in reality means that there are 20 fill variables:
xFill[2], ..., xFill [20].
```

For each assignment made to an arrayed fill variable it ideally should be to each of the array elements:

```
xFill[i] := 'this' means:

xFill[1] := 'this', xFill[2] := 'this', ..., Fill[20' :=
'this'
```

However, when displayed (ie, ^xFill[i]) the display should be the unique values relevant to the index, such as "[this/that]."

Assignments made to explicit array items should have unique names if the arrayed items are never sorted, chosen at random, or otherwise used in a manner that would make programming problematic.

Avoid this:

```
xFill[1] := 'this'
xFill[2] := 'that'
xFill[3] := 'these'
xFill[4] := 'those'
```

Preferred Method:

```
IF A1 = Current THEN
       xthis_that := 'this'
       xthese_those := 'these'

ELSE
       xthis_that := 'that'
       xthese_those := 'those'
ENDIF
```


Program Format - Answer Names and the ORD Function in the Rules

It is preferable not to use the ORD function in program code in the following manner:

```
IF Married.ORD = 1 THEN

    SpouseName
    SpouseBirthDate

ENDIF
```

Use the more readable "normal" programming method:

```
IF Married = Yes THEN

    SpouseName
    SpouseBirthDate

ENDIF
```

In reality, the ORD function should only be used when making an assignment to an enumerated field from a different enumerated field, or passing an enumerated to an integer field.

Use caution, though: if you do not have a code value zero defined in your destination field, you will get an "imputation error" if there's a DK/RF/EMPTY value in the source field.

```
MyRelation := SomeRelation.ORD
```

Consistency Checks

Be redundant and use the keyword CHECK or SIGNAL before hard and soft consistency checks. Indent using the following guidelines:

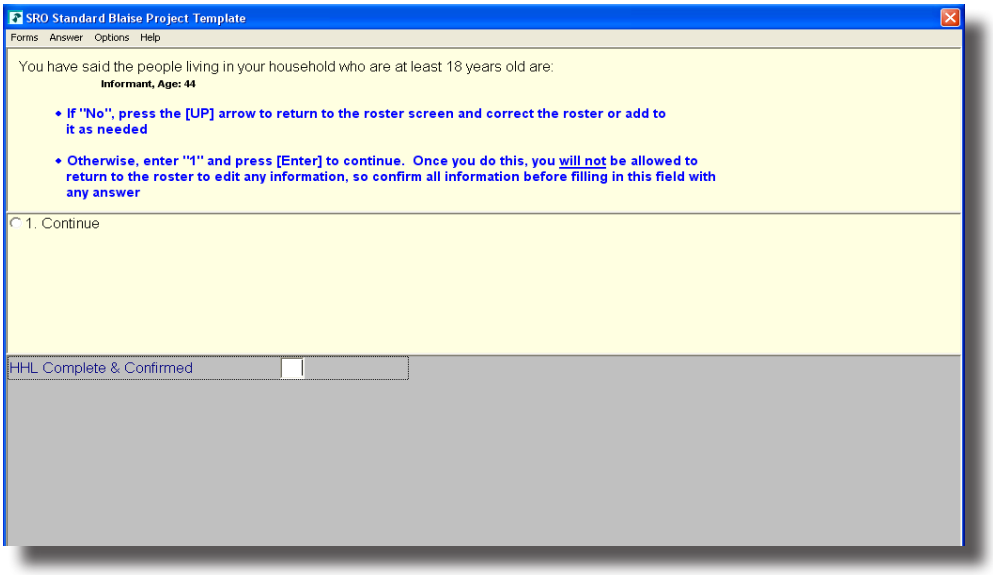
```
CHECK
       B20.YEAR > 1985 OR B20 = NONRESPONSE
       "@D@Rw@R@D @R@BCheck B20.Year@B@R
       @/@/@/|@D@Rw@R@D @R@BThe birth year must be later
       than 1985@B@R"
```

```
IF B20.Month = 2 AND B20.Day = 29 THEN
        SIGNAL
                B20.Day < 28 OR B20 = NONRESPONSE
                "@Dw@D @ISignal B20.Day@I
                @/@/@|@Dw@D @IUsually only 28 days in
                February; is the 29th true?@I"
        ENDIF
```


HHL Rosters

Appendix C contains sample code for a basic HHL.

Most panel studies, will have a "confirmation" screen after the roster. The interviewer must endorse the roster verifying that the correct Respondent has been selected and as an indicator to move ahead and copy information from the roster downstream to other blocks and fields if necessary.



Appendices

Appendix A • Quick Reference

Quick Reference

This section will give users an "at-a-glance" view of four representations of Blaise CAI instrument developed using SRO standards and tools, as reflected in specific question examples.

The following 24 examples cover typical question types, and show question specifications, programming code, and the resulting Blaise Data Entry Program (DEP) screen and Michigan Questionnaire Documentation System (MQDS) output.

The goal of this section is to limit ambiguity about what is specified, what is programmed, and how these items will appear to end-users. The objective is to develop consistently specified and programmed instruments that result in consistent screen displays and instrument documentation.

Note that on several of the pages there is boxed text that is included to help describe shortcuts that the programmer used. This text is included only as a reference to help non-programmers get a better idea of what the code means. These text boxes look like this:

TYesNo : (Yes (1), No (5))

In this example, the programmer has used reusable short programming code, "TYesNo," as an economic way to represent more detailed code for response options that only had to be programmed once.

Yes/No with Skip Logic

Specification

B1
Do you currently use a computer, either at work, at home, or at school?

/ "Currently Use a Computer"

Yes	1	
No	5	GO TO B3
DK		GO TO B3
RF		GO TO B3

B2
Have you used the Internet

/ "Used the Internet"

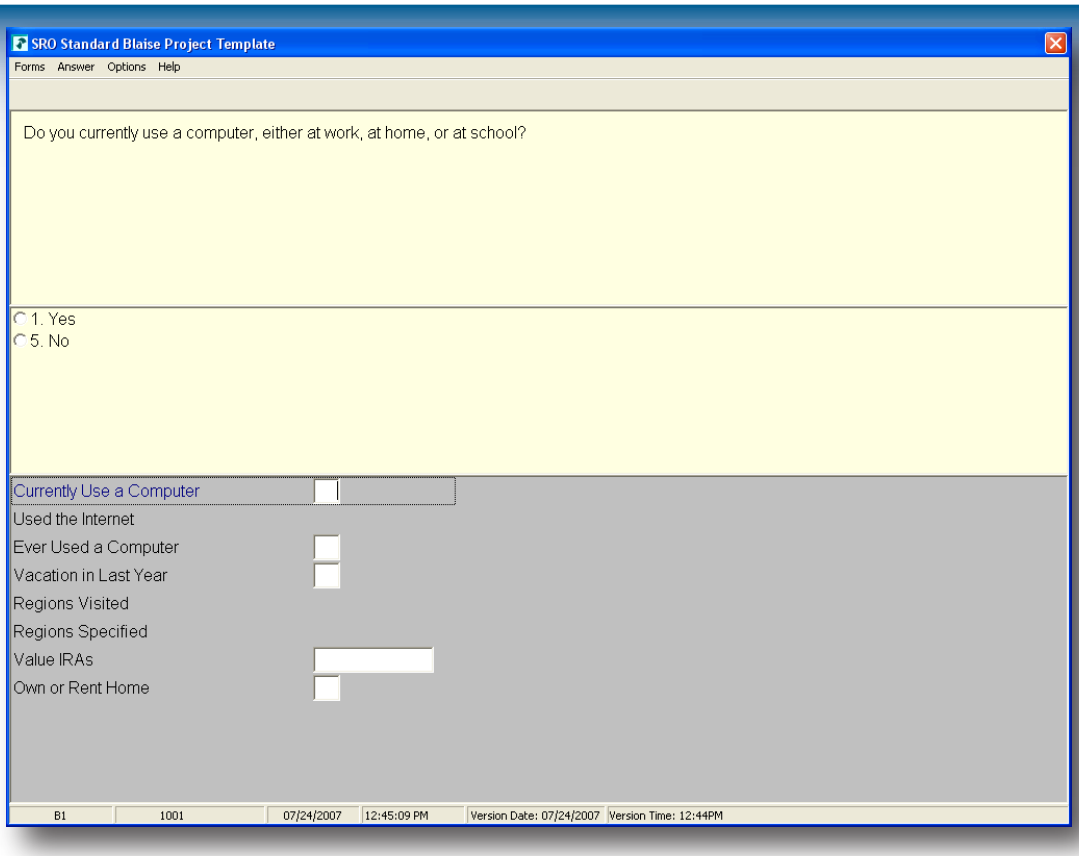
Yes	1	GO TO B4
No	5	GO TO B4
DK		GO TO B4
RF		GO TO B4

Programming

```
B1      (B1)
        "Do you currently use a computer, either at work, at home, or
        at school?" /
        "Currently Use a Computer" :

        TYesNo
```

TYesNo :
(Yes (1),
No (5))

Blaise DEP



QR_B1.ai

MQDS

SECTION_B

B1

Do you currently use a computer, either at work, at home, or at school?

- 1 Yes
- 5 No GOTO B3

Enumerated Response Short List

B9

(B9_Internet)

There is a lot of talk these days about the Internet. Some people have heard or read about the Internet, while others have not. How about you -- do you know a fair amount about the Internet, have you heard about the Internet, but don't know much about it, or have you never heard of the Internet?

/ "Internet Awareness"

Know a fair amount	1
Heard about it, but don't know much	2
Never heard of it	3
DK	
RF	

Specification

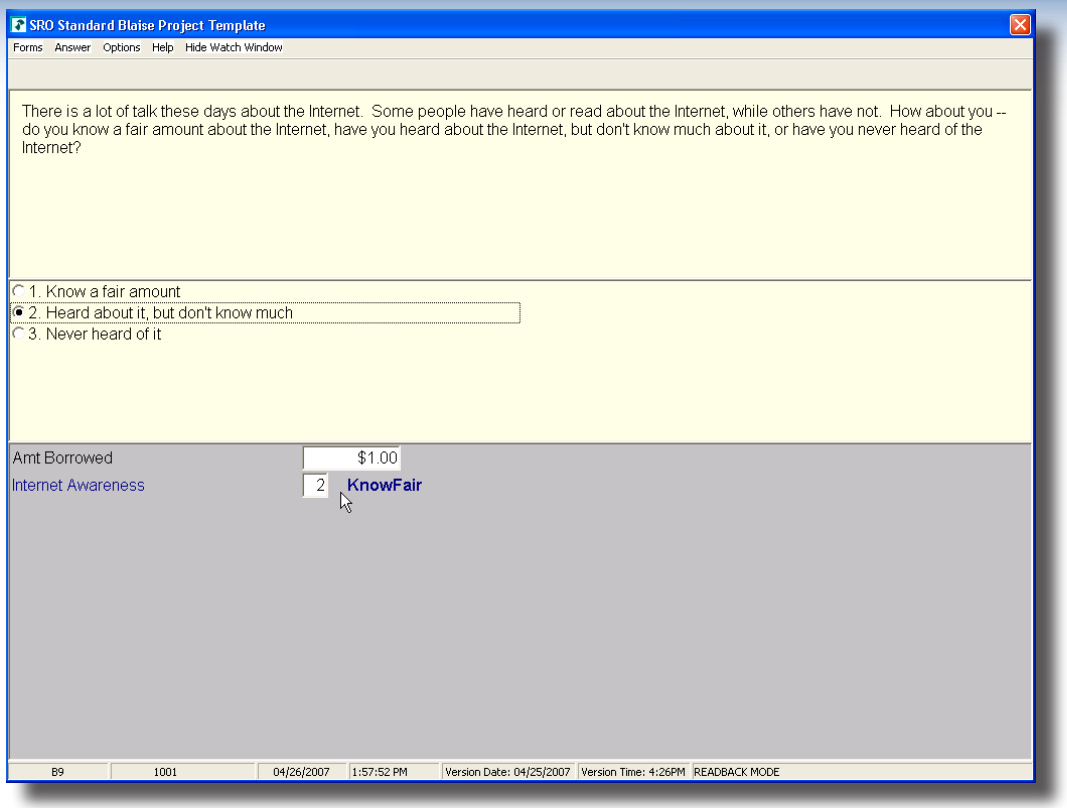
```
B9      (B9_Internet)
        There is a lot of talk these days about the Internet.••Some
        people have heard or read about the Internet, while others have
        not.••How about you -- do you know a fair amount about the In-
        ternet, have you heard about the Internet, but don't know much
        about it, or have you never heard of the Internet?/
        "Internet Awareness" :

        TKnowledge
```

TKnowledge:

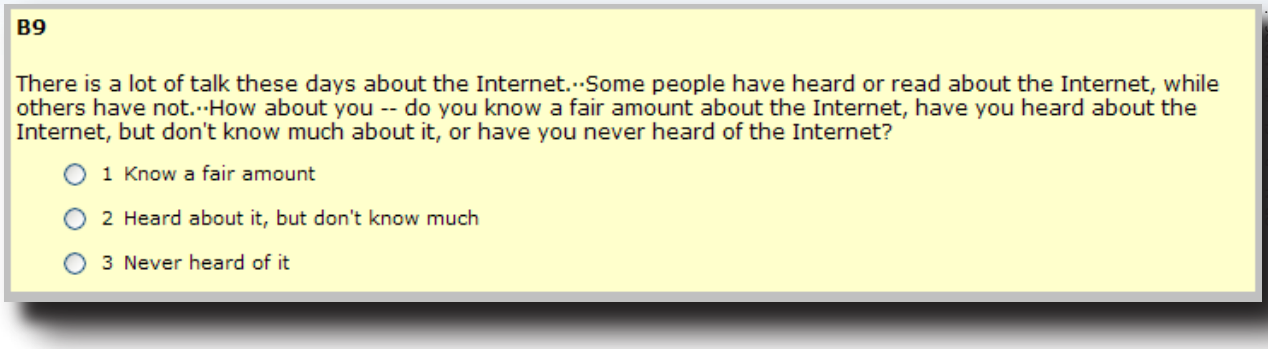
(KnowFair	*(1) "Know a fair amount",*
HeardDK	*(2) "Heard about it, but don't know much",*
Never	*(3) "Never heard of it")*

Programming

Blaise DEP



MQDS



Introduction Question

Intro
Hello, my name is [Iwer Name] and I am calling from The University of Michigan...

- Provide a description of The University of Michigan if R asks
- ENTER [1] to continue

/"Introduction"

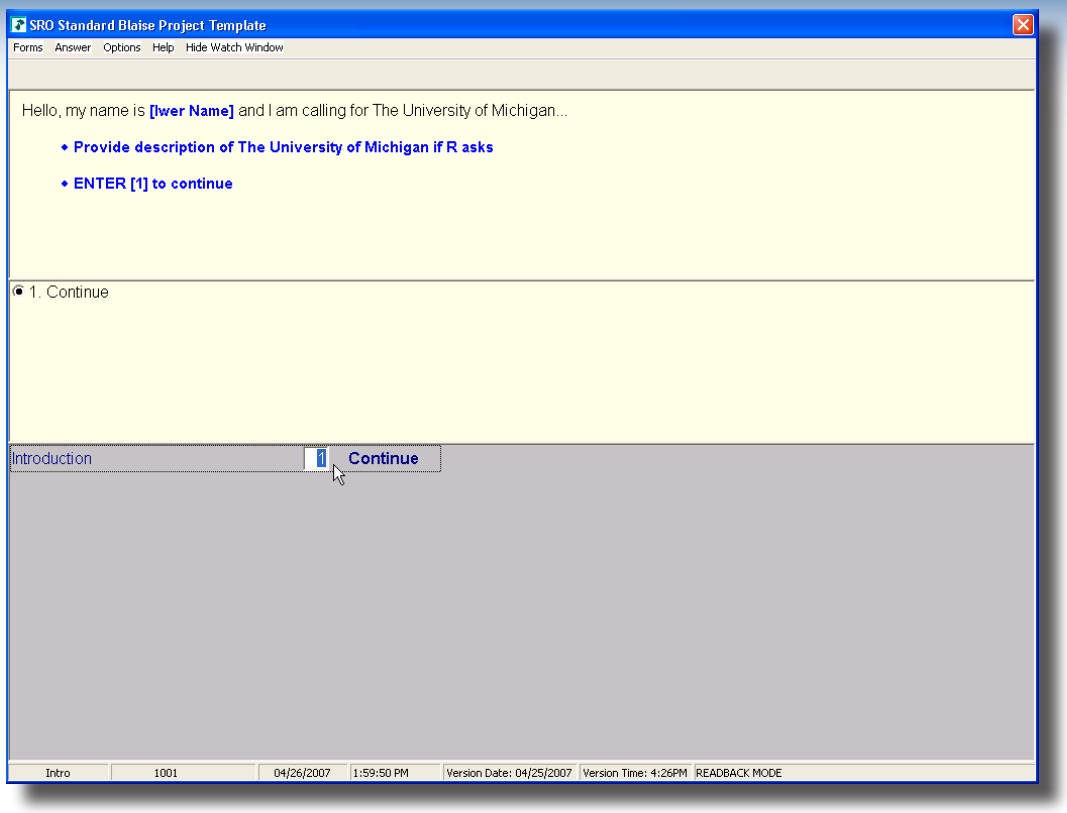
Specification

```
Intro (Intro)

        "Hello, my name is @I[Iwer Name]@I and I am calling for The
        University of Michigan...

        @/@/@|@Dw@D @IProvide description of The University of Michi-
        gan if R asks@I

        @/@/@|@Dw@D @IENTER [1] to continue@I" /
        "Introduction" :

        TContinue
```

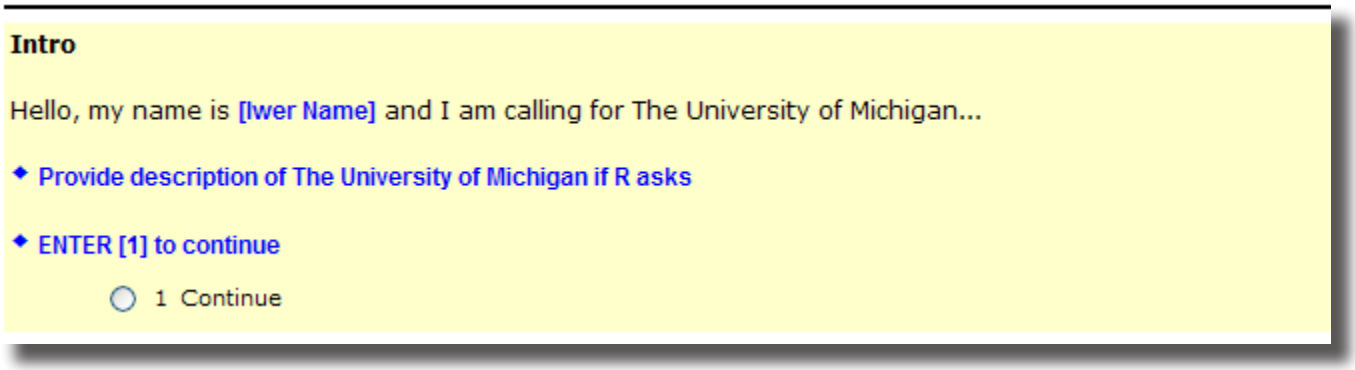
TContinue:
(Continue)

Programming

Blaise DEP



MQDS



Fixed Length String

B11

In what city and state was your mother living when you were born?

City: _____

State: _____

/"Birth City"

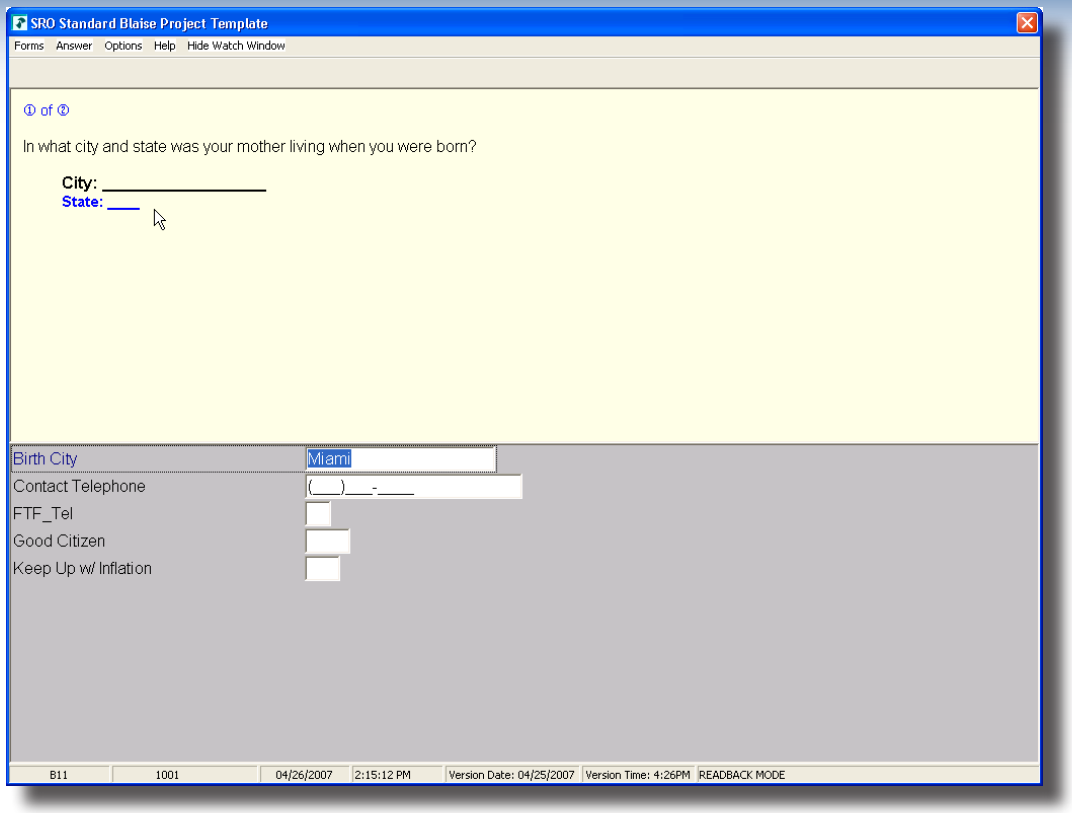
String; width = 20

Specification

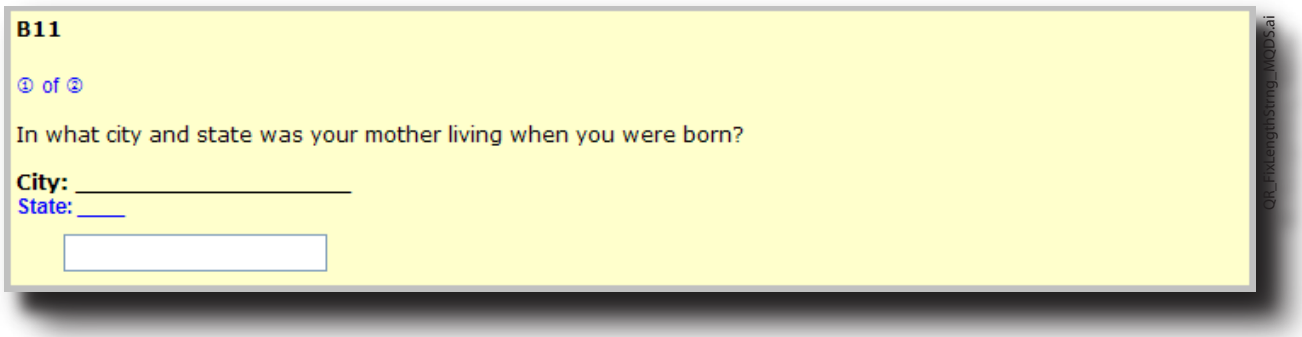
```
B11      (B11)
         "@Ej@E @Wof@W @Ek@E
         @/@/In what city and state was your mother living when you were
         born?

         @/@/@|@BCity: _____@B
         @/@|@IState: ____ @I" /
         "Birth City" :

         STRING[20]
```

Programming

Blaise DEP



MQDS



Open Ended

B13

People have different ideas about what being a good citizen means. We're interested in what you think. How would you describe a good citizen in this country -- that is, what things about a person are most important in showing that one is a good citizen? (Any other ideas?)

/"Good Citizen"

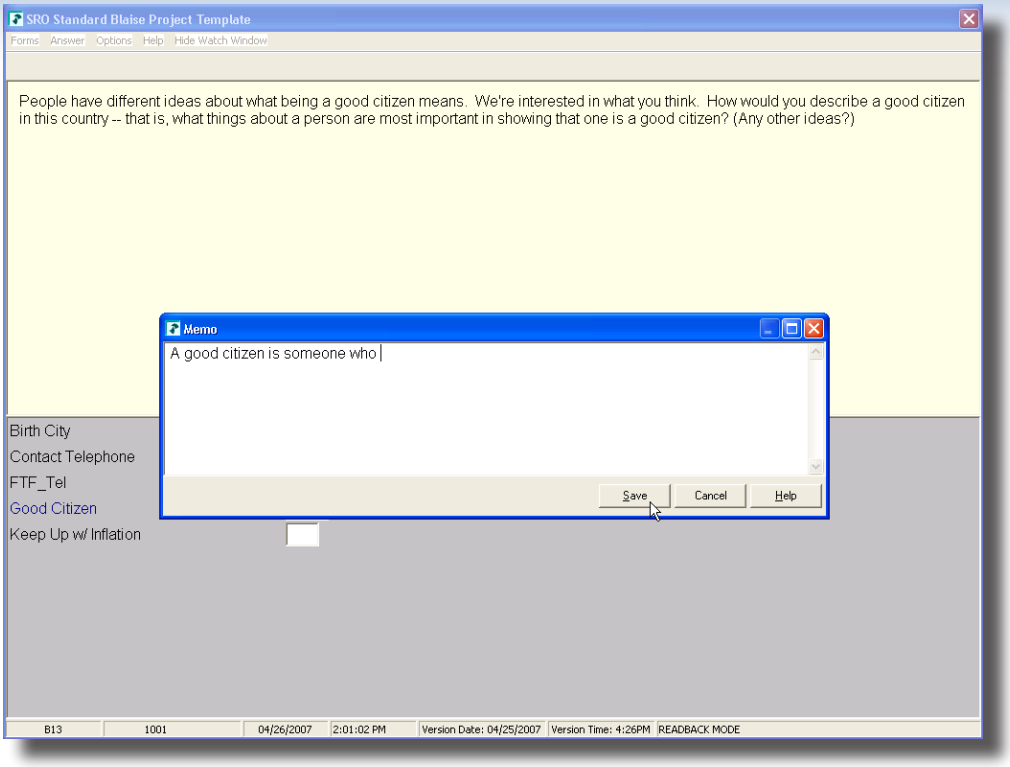
Open

Specification

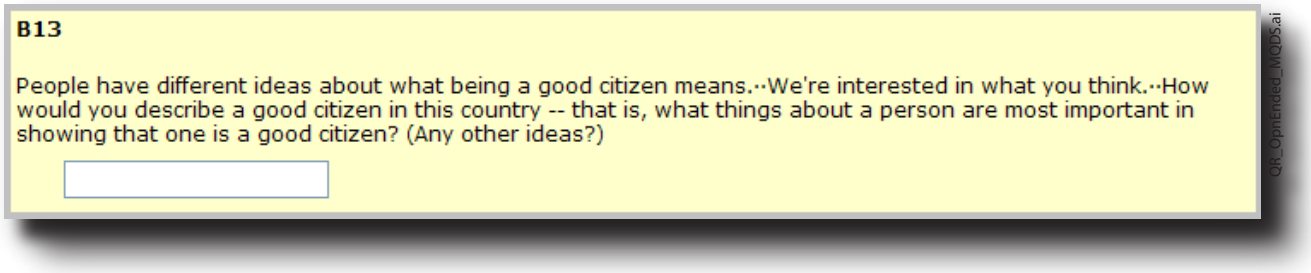
```
B13      (B13)
         "People have different ideas about what being a good citi-
         zen means.••We're interested in what you think.••How would
         you describe a good citizen in this country -- that is, what
         things about a person are most important in showing that one
         is a good citizen? (Any other ideas?)" /
         "Good Citizen" :

         OPEN
```

Programming

Blaise DEP



MQDS



Whole Number

B14
RB Page 3

On a scale of 0 to 100, what do you think are the chance that your income will keep up with inflation for the next five years?

/"Keep Up with Inflation"

Integer; range 0-100

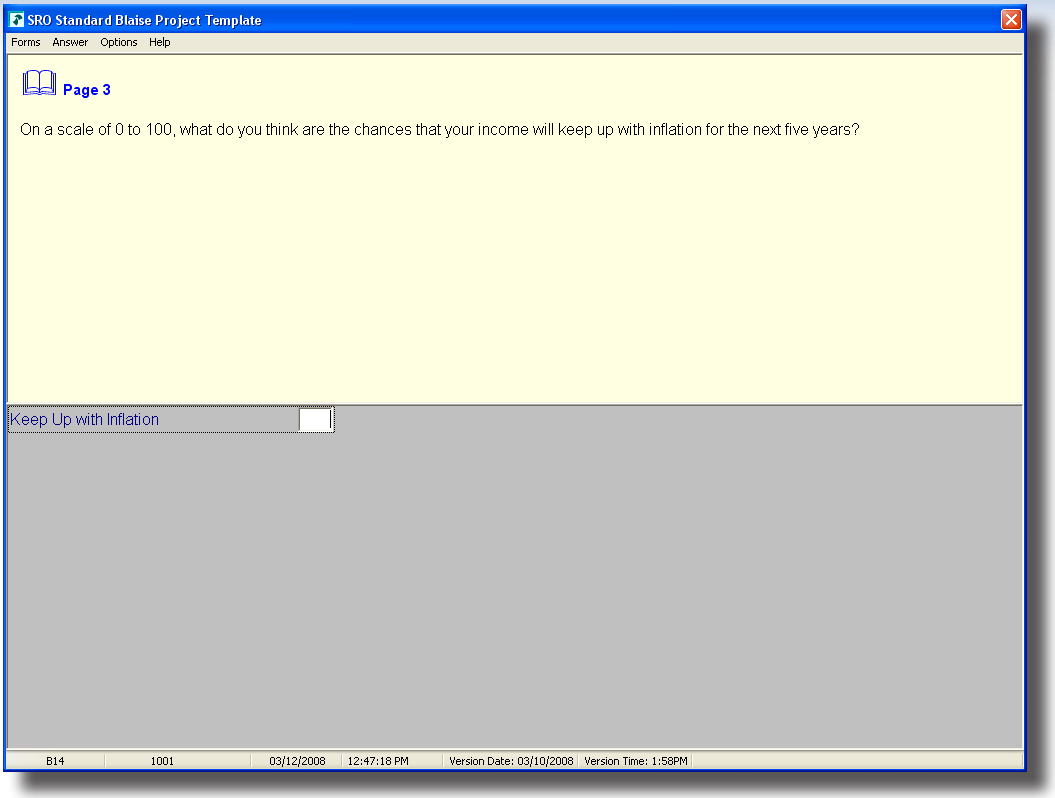
Specification

```
B14     (B14)
        "@F&@F @IPage 3@I
        @/@/On a scale of 0 to 100, what do you think are the chanc-
        es that your income will keep up with inflation for the next
        five years?" /
        "Keep Up with Inflation" :

        TRange0_100          {0..100}
```

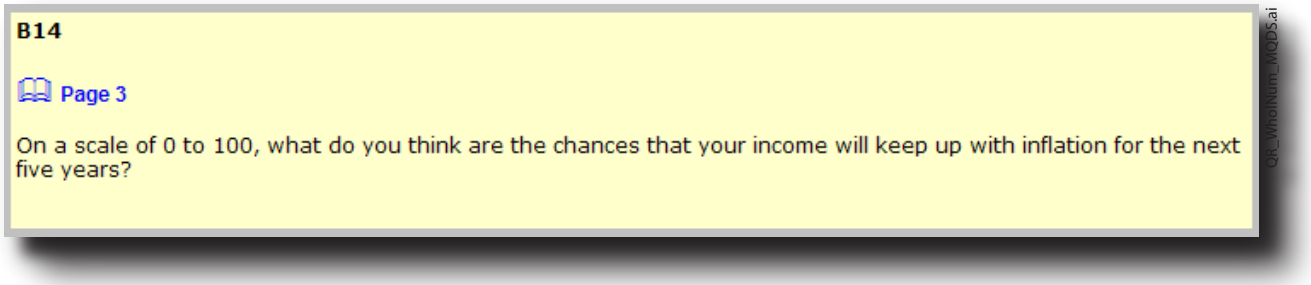
TRange0_100 :
0..100

Programming

Blaise DEP



MQDS



Whole Number with Decimals

B15a

Specification

How much do you earn now from this job?

$ _____

/"Earnings Amount"

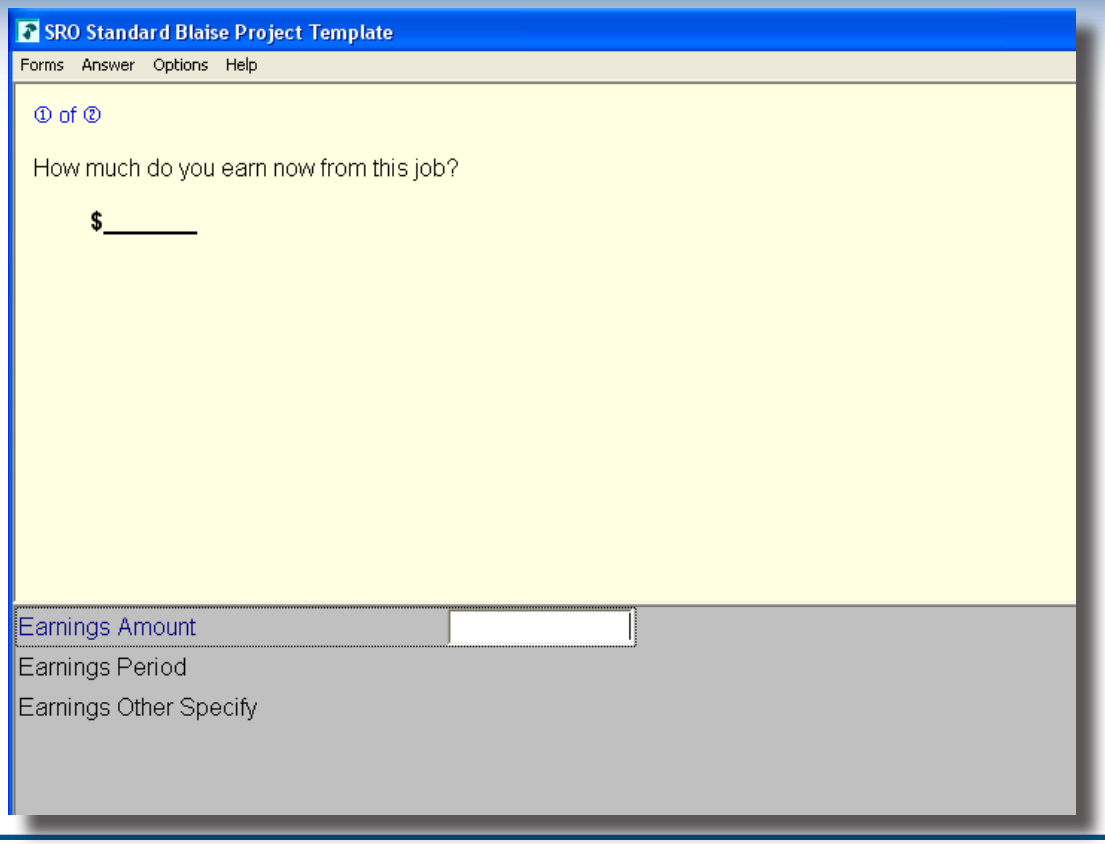
Currency; range 1.00 - 99,999,999.00

Programming

```
B15a    (B15a)
        "@Ej@E @Wof@W @Ek@E
        @/@/How much do you earn now from this job?

        @/@/@|@B$_______@B" /
        "Earnings Amount" :

        TDollar1_99999999_00            {1.00..99,999,999.00}
```

TDollar1_99999999_00:
1.00..99999999.00

Blaise DEP



MQDS

B15a

① of ②

How much do you earn now from this job?

$_____

1 - 99,999,999

Hard Consistency Check (Check)

Specification

B25
"How many times in the past year have you been drunk on alcohol to the point of passing or blacking out?

- If Respondent says "never", enter [96]
- If Respondent says "DK", probe: Would you say, 5, 10, or more than 10 times?
- IF Respondent says more than 10 times, enter [97]

/"Freq of Passing Out Drunk" :

Integer, 1-10, 96, 97

==========

B26
"How many times in the past year have you become ill from drinking?

- If Respondent says "never", enter [96]
- If Respondent says "DK", probe: Would you say, 5, 10, or more than 10 times?
- IF Respondent says more than 10 times, enter [97]

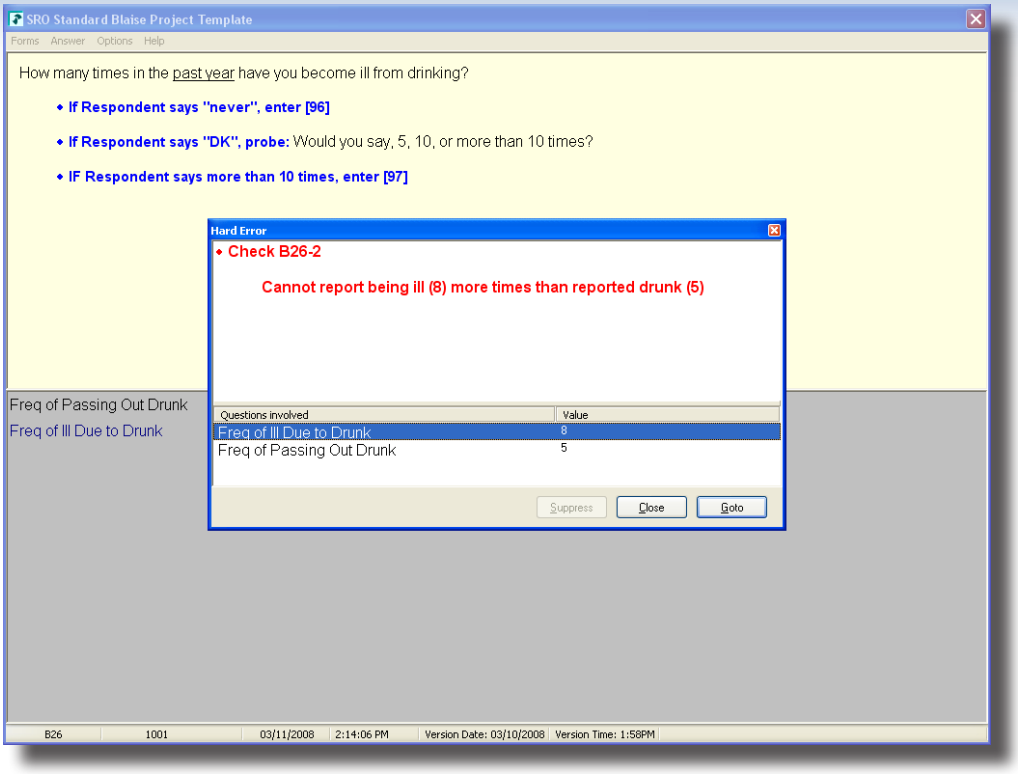
/ "Freq of Ill Due to Drunk" :

Integer, Range 1-10, 96, 97

Programming

```
B25        (B25)
        "How many times in the @Upast year@U have you been drunk on alcohol to the point of passing or
        blacking out?
        @/@/@|@Dw@D @IIf Respondent says ""never"", enter [96]@I
        @/@/@|@Dw@D @IIf Respondent says ""DK"", probe:@I Would you say, 5, 10, or more than 10 times?
        @/@/@|@Dw@D @IIF Respondent says more than 10 times, enter [97]@I" /
        "Freq of Passing Out Drunk" :

        TRange1_97 {1..10, 96, 97}

B26        (B26)
        "How many times in the @Upast year@U have you become ill from drinking?
        @/@/@|@Dw@D @IIf Respondent says ""never"", enter [96]@I
        @/@/@|@Dw@D @IIf Respondent says ""DK"", probe:@I Would you say, 5, 10, or more than 10 times?
        @/@/@|@Dw@D @IIF Respondent says more than 10 times, enter [97]@I" /
        "Freq of Ill Due to Drunk" :

        TRange1_97 {1..10, 96, 97}

RULES
        B25
        IF B25 = RESPONSE THEN
            CHECK
               B25 <= 10 OR B25 = 96 OR B25 = 97
               "@D@Rw@R@D @R@BInvalid response, please re-enter@B@R"
        ENDIF
        B26
        IF B26 = RESPONSE THEN
            CHECK
                B26 <= 10 OR B26 = 96 OR B26 = 97
                "@D@Rw@R@D @R@BInvalid response, please re-enter@B@R"
            CHECK
                B26 <= B25 OR B26 = 96 OR B26 = 97
                "@D@Rw@R@D @R@BCannot report being ill (^B26) more times than reported drunk (^B25)@B@R"
        ENDIF
```

Blaise DEP



MQDS

B26

How many times in the past year have you become ill from drinking?

- If Respondent says "never", enter [96]
- If Respondent says "DK", probe: Would you say, 5, 10, or more than 10 times?
- IF Respondent says more than 10 times, enter [97]

1 - 97 GOTO IWComplete

HARD Edit

Valid condition: ((B26 <= 10) OR (B26 = 96)) OR (B26 = 97.00000000000001)

Error returned to the user: • Check B26-1

Invalid response, please re-enter

Involved fields:

HARD Edit

Valid condition: ((B26 <= B25) OR (B26 = 96)) OR (B26 = 97.00000000000001)

Error returned to the user: • Check B26-2

Cannot report being ill ([B26]) more times than reported drunk ([B25])

Involved fields:

Soft Consistency Check (Signal)

B24.Age
How old is [Child Name]?

/"Child Age"

Integer; range: 1-20

Soft Consistency Check: Child's age must be less than R's age - 15;
"Age is within 15 years of R's age (Display R's Age)
Does this mean you were ^xRAgeMinus15 years old when this child was born?

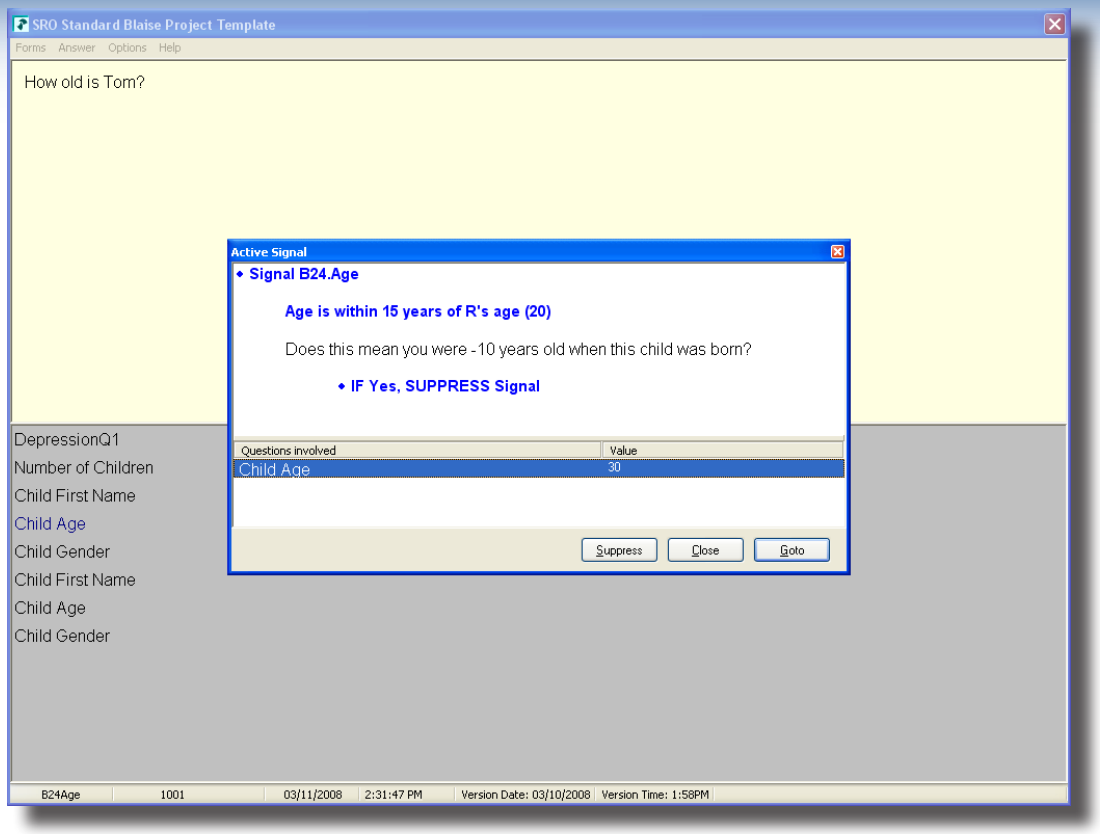
- IF Yes, SUPPRESS Signal"

Specification

```
Age     (B24Age)
         "How old is ^Name?" /
         "Child Age" :

         TAge

RULES
     IF Age = RESPONSE THEN
            IF Age >= (xRAge - 15) THEN
                  xRAgeMinus15.KEEP
                   xRAgeMinus15 := (xRAge - Age)
                   SIGNAL
                        Age <= (xRAge - 15)
                         INVOLVING (Age)
                         "@Dw@D@ISignalB24.Age@I
                         @/@/@|@IAge is within 15 years of R's age (^xRAge)@I
                         @/@/@|Does this mean you were ^xRAgeMinus15 years old
                  when this child was born?
                         @/@/@|@|@Dw@D @IIF Yes, SUPPRESS Signal@I"
             ENDIF
       ENDIF
```

Programming

Blaise DEP



MQDS

Age

How old is [NAME]?

0 - 99 GOTO Gender

SOFT Edit

Valid condition: (Age <= (xRAge - 15)) INVOLVING (Age)

Error returned to the user: Signal B24.Age

Age is within 15 years of R's age ([AGE B20])

Does this mean you were [xRAge - Age] years old when this child was born?

Multiple Responses

B10_
RB Page 1

Which of the following best describes your race or ethnic origin: White, Black or African-American, Hispanic, Native Hawaiian, American Indian, Alaskan Native, Asian, or Pacific Islander?

- PROBE before accepting refusal
- ENTER all that apply
- For multiple responses, use [Space] or [-] to separate responses

/"Race/Ethnicity"

White	1
Black/African-American	2
Hispanic	3
Native Hawaiian	4
American Indian	5
Alaskan Native	6
Asian	7
Pacific Islander	8
Other - specify	97

Specification

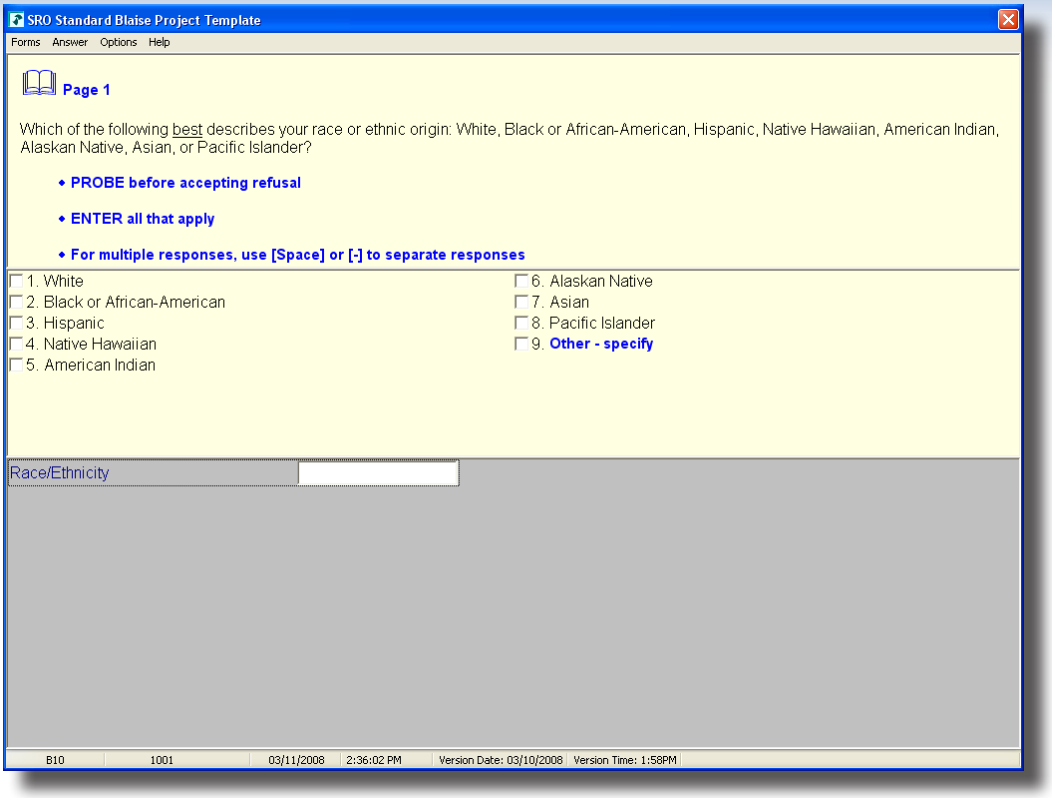

```
B10_    (B10_)
        @F&@F @IPage 1@I
        @/@/Which of the following @Ubest@U describes your race or
        ethnic origin: White, Black or African-American, Hispanic,
        Native Hawaiian, American Indian, Alaskan Native, Asian, or
        Pacific Islander?

        @/@/@|@Dw@D @IPROBE before accepting refusal@I
        @/@/@|@Dw@D @IENTER all that apply@I
        @/@/@|@Dw@D @IFor multiple response, use [Space] or [-] to
        separate responses@I" /
        "Race/Ethnicity" :

        SET OF TRace
```

TRace:

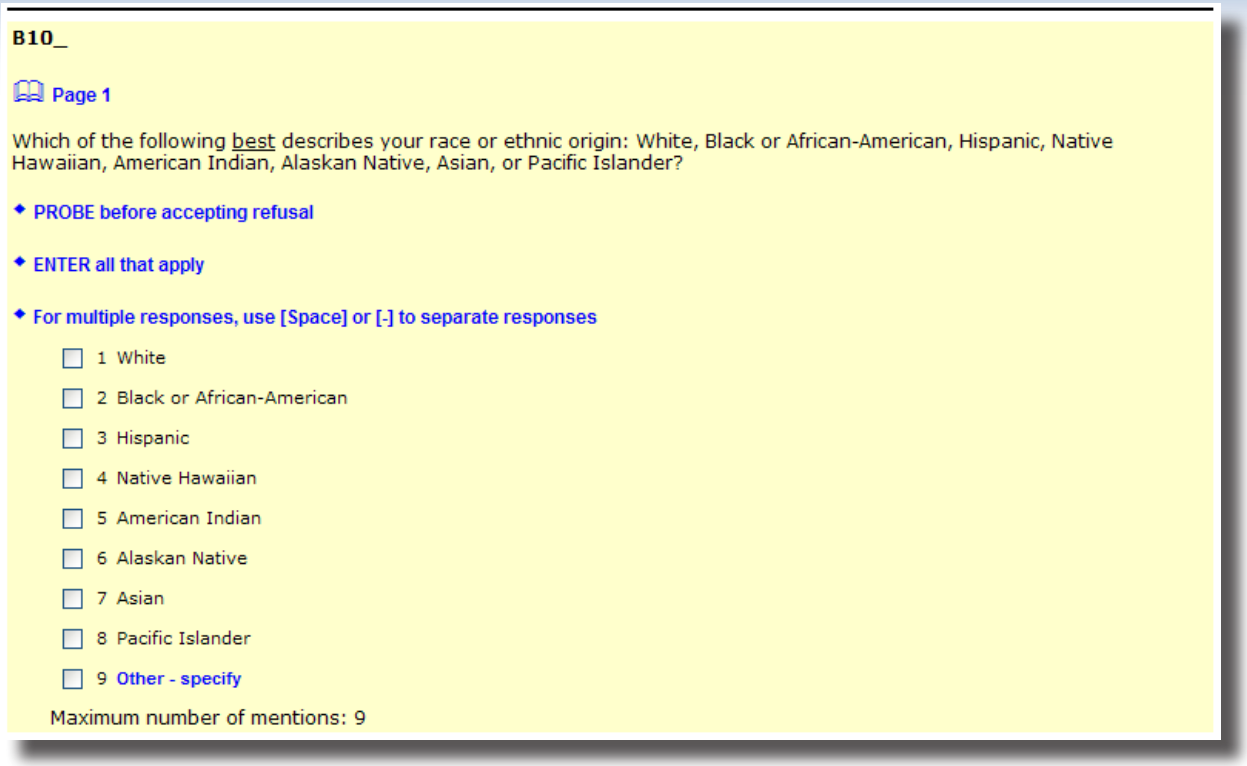
(White	*(1)*	*"White",*
Black	*(2)*	*"Black or African-American",*
Hispanic	*(3)*	*"Hispanic",*
NativHaw	*(4)*	*"Native Hawaiian",*
AmerIndian	*(5)*	*"American Indian",*
AlaskNat	*(6)*	*"Alaskan Native",*
Asian	*(7)*	*"Asian",*
PacIsland	*(8)*	*"Pacific Islander",*
Other	*(9)*	*"@IOther - specify@I")*

Programming

Blaise DEP



MQDS



Entry Mask

B12

Is there an additional telephone number where we may reach you if this number fails in the future?

- If the Respondent mentions a cell phone number, please enter that number and then enter a [F2] note and indicate that it is a cell number
- If the Respondent offers a relatives' or friends' number, do not enter that number. Explain that we can only use numbers that they "own" for this section

/"Contact Telephone"

String; Telephone Mask, width = 10

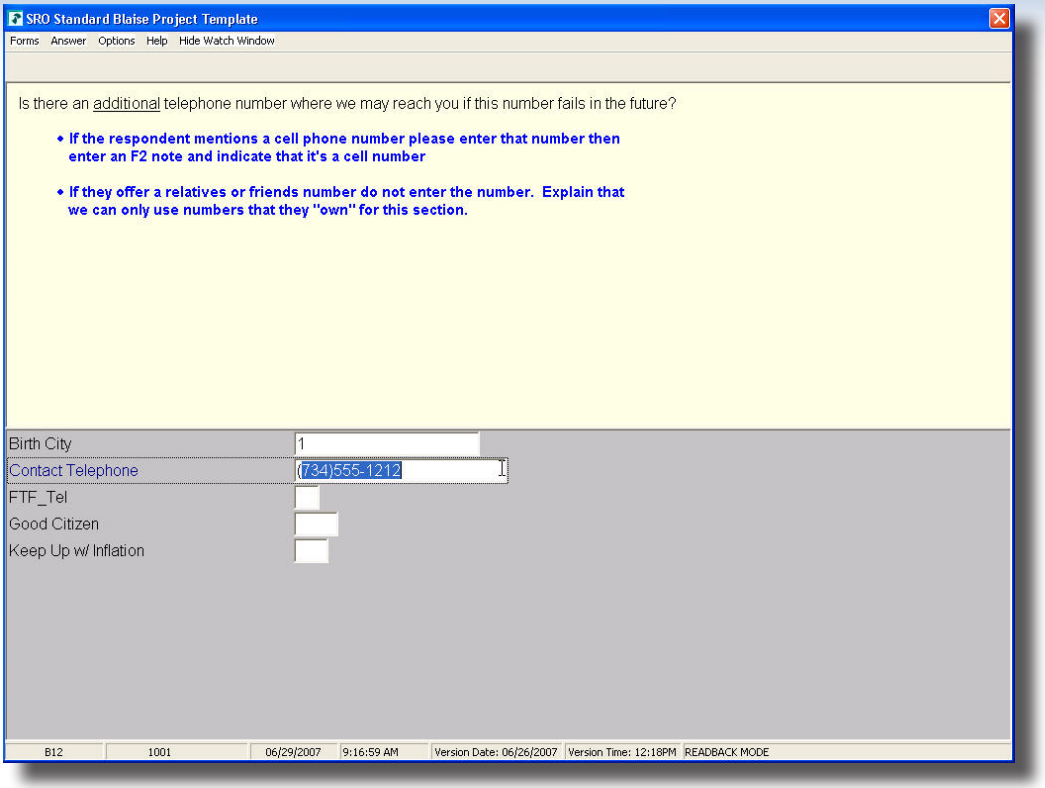
Specification

```
B12     (B12)
        "Is there an @Uadditional@U telephone number where we may
        reach you if this number fails in the future?

        @/@/@|@Dw@D @IIf the respondent mentions a cell phone num-
        ber please

        @/@|•••enter that number then enter an F2 note and indicate
        that it's a cell number@I

        @/@/@|@Dw@D @IIf they offer a relatives or friends number
        do not enter the

        @/@|•••number.••Explain that we can only use numbers that
        they ""own"" for this section." /
        "Contact Telephone" :

        TPhone
```

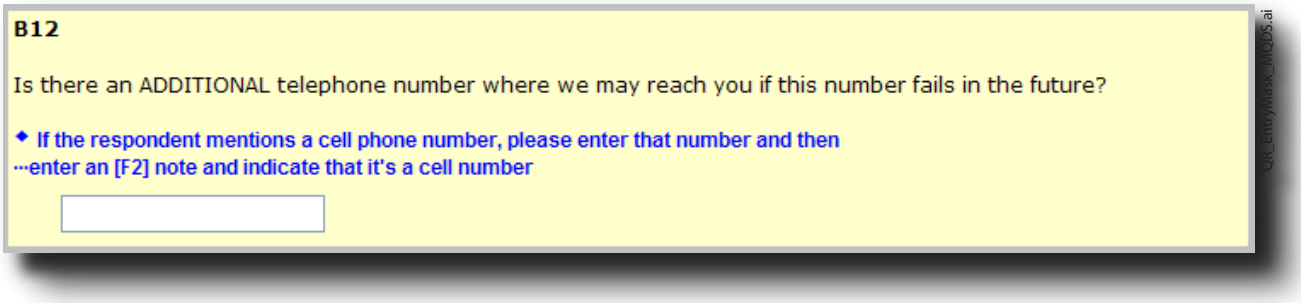
TPhone:
STRING[10]

Programming

Blaise DEP



MQDS



Multipart Questions

B15a
① of ②

How much do you earn now from this job?

$_____

/"Earnings Amount"

Question Type: Currency, Range 1.00 - 99,999,999.00
/ "Earnings Amount"

B15b
② of ②

(How much do you earn now from this job?)

$_____ per (Hour/Week/Two weeks/Month/Year)

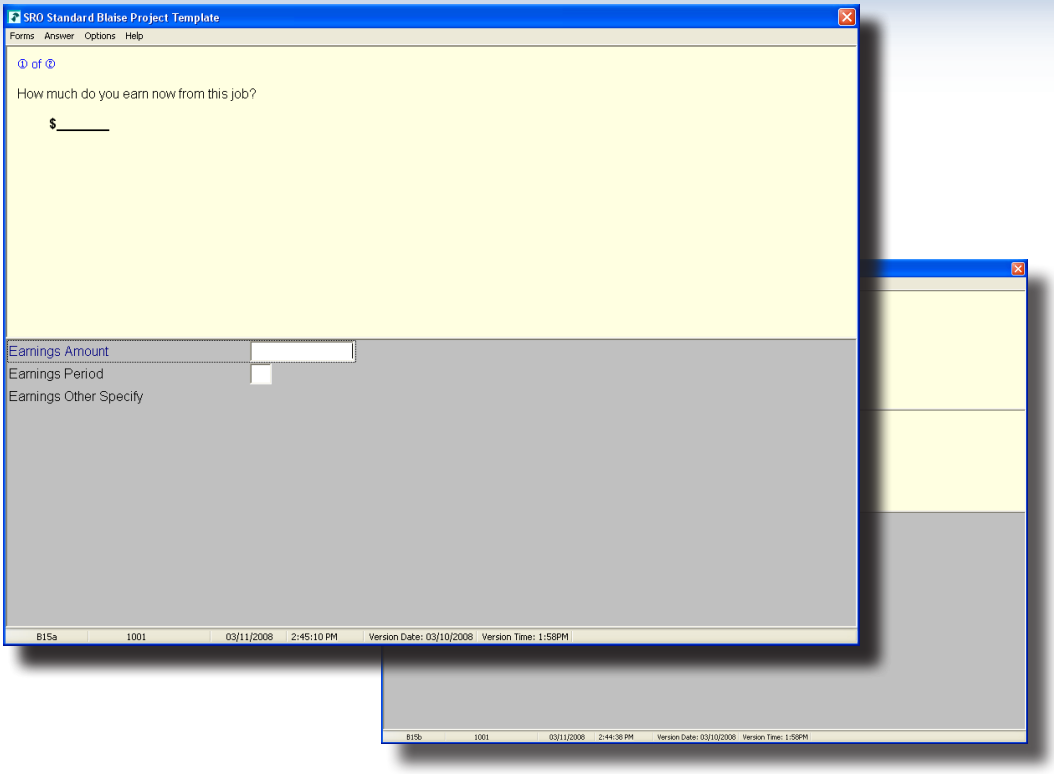
/"Earnings Period"

Hour	1
Week	2
Two weeks	3
Month	4
Year	5
Other - specify	6


```
B15a    (B15a)
        "@Ej@E @Wof@W @Ek@E
        @/@/How much do you earn now from this job?

        @/@/@|@B$_______@B" /
        "Earnings Amount" :

        TDollar1_99999999_00            {1.00..99,999,999.00}
```

Programming

Blaise DEP



MQDS

B15a

① of ②

How much do you earn now from this job?

$_____

1 - 99,999,999

B15B_IC_12

B15a > 0

○ 1 EXPR IS FALSE GOTO B18a

○ 2 EXPR IS TRUE

B15b

② of ②

(How much do you earn now from this job?)

[B15A] per **(Hour/Week/Two weeks/Month/Year)**

○ 1 Hour GOTO B18a

○ 2 Week GOTO B18a

○ 3 Two weeks GOTO B18a

○ 4 Month GOTO B18a

○ 5 Year GOTO B18a

○ 6 **Other -- specify**

Scale Questions - All Points

B18a
RB Page 3

Specification

How much do you agree with the following statement:

Patriotism is an important part of being a good citizen.

Do you strongly agree, agree, somewhat agree, neither agree or disagree, somewhat disagree, disagree, or strongly disagree?

/"Patriotism Important"

Strongly agree	1
Agree	2
Somewhat agree	3
Neither agree or disagree	4
Somewhat disagree	5
Disagree	6
Strongly disagree	7

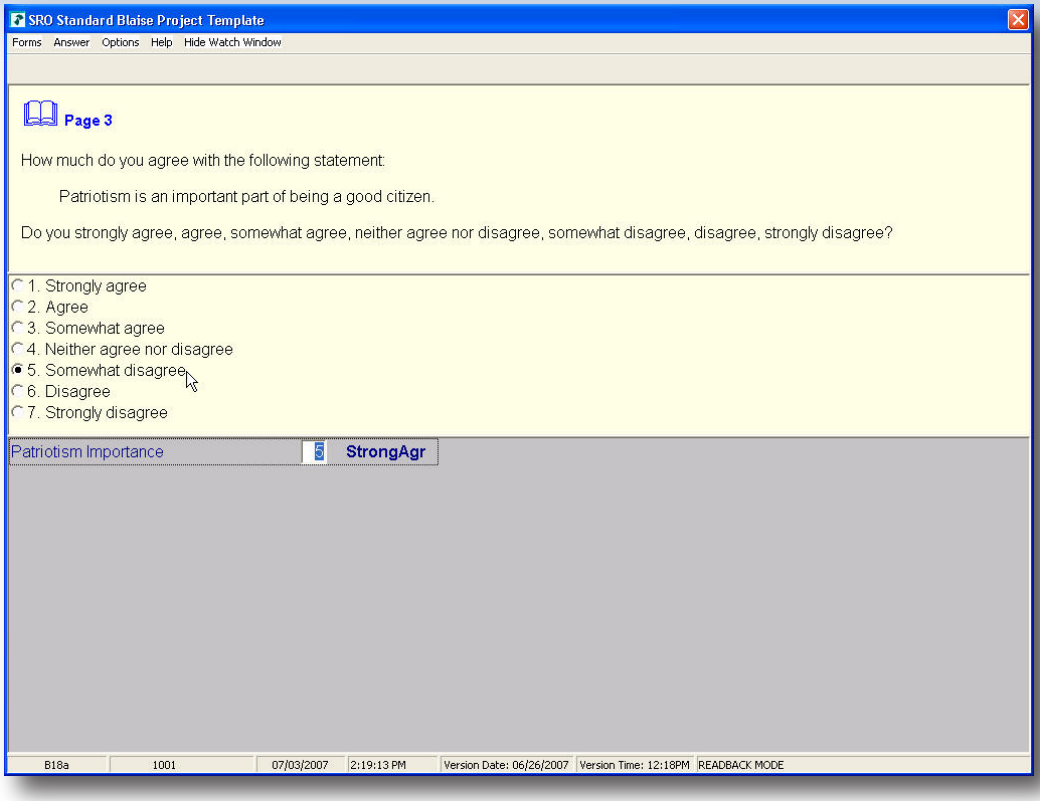
Programming

```
B18a    (B18a)
        "@F&@F @IPage 3@I
        @/@/How much do you agree with the following statement:

        @/@/@|""Patriotism is an important part of being a good cit-
        izen.""

        @/@/Do you strongly agree, agree, somewhat agree, neither
        agree nor disagree, somewhat disagree, disagree, strongly
        disagree?" /
        "Patriotism Importance" :

        TScale1_7Agree_a
```

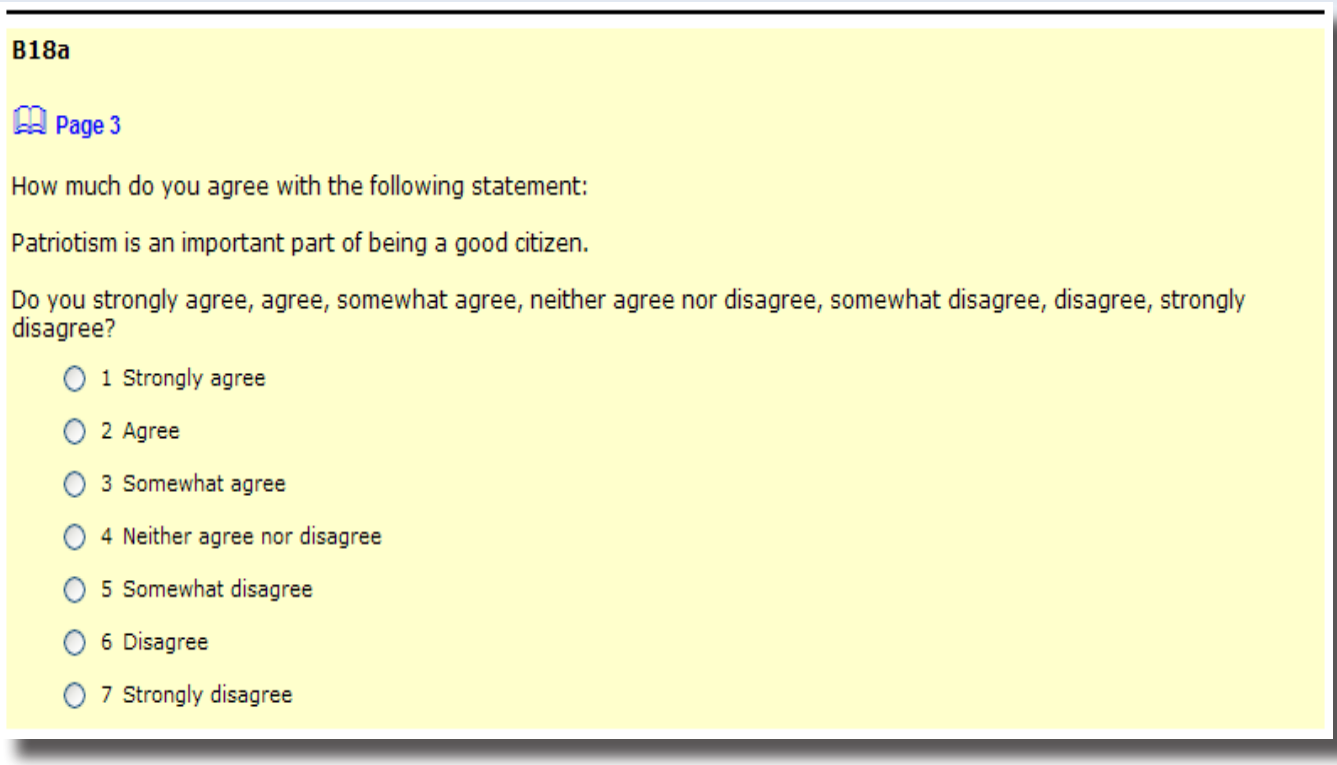
TScale1_7Agree_a:

(Strng_Agr	*(1)*	*"Strongly Agree",*
Agr	*(2)*	*"Agree",*
SWhat_Agr	*(3)*	*"Somewhat Agree",*
Neither_A_D	*(4)*	*"Neither Agree or Disagree",*
SWhat_Dis	*(5)*	*"Somewhat Disagree",*
Dis	*(6)*	*"Disagree",*
Strng_Dis	*(7)*	*"Strongly Disagree")*

Blaise DEP



MQDS



Interviewer Checkpoint On Screen

B12a

☑ Interviewer Checkpoint

- Is this a Face to Face interview or Phone?

/"Iwer Checkpoint FTF_Tel"

Face to face	1
Telephone	2

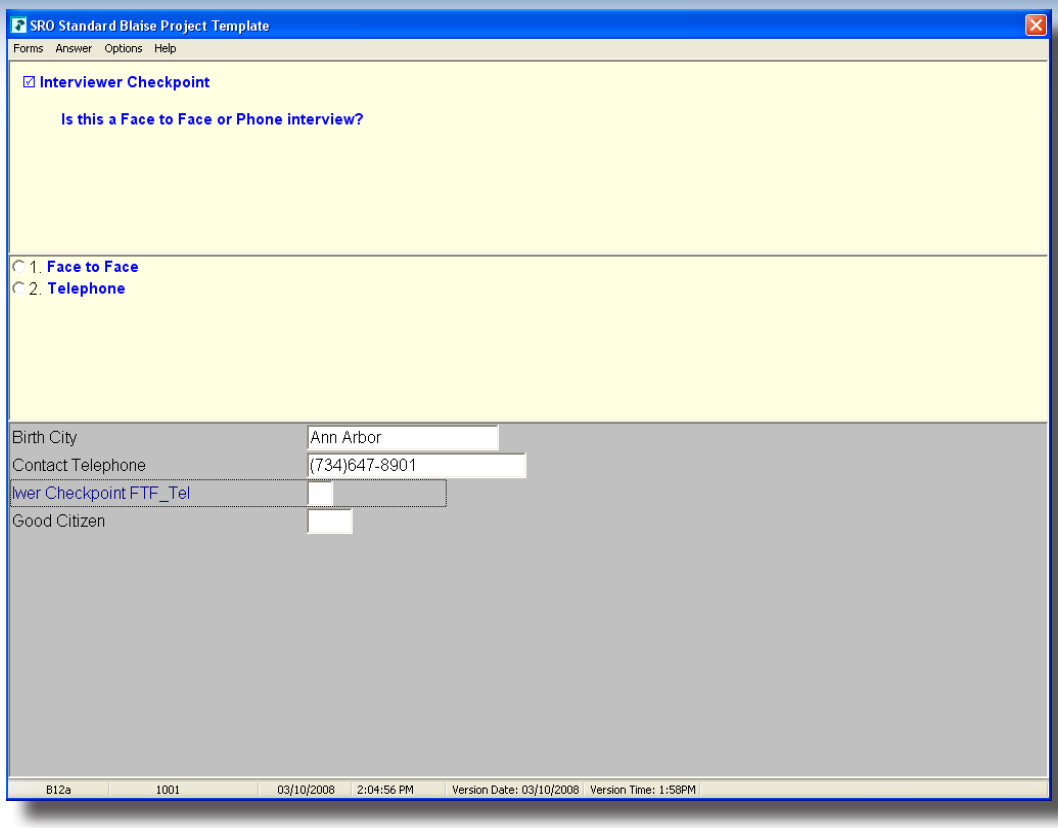
NO DK, NO RF

Specification


```
B12a    (B12a)
        "@ER@E @IInterviewer Checkpoint@I
        @/@/@|@Dw@D @IIs this a Face to Face interview or
        Phone?@I"/
        "FTF_Tel" :

        TFTF_Tel
```

Programming

Blaise DEP



MQDS

Interviewer Instructions

B12
Is there an additional telephone number where we may reach you if this number fails in the future?

- If the respondent mentions a cell phone number please enter that number then enter an F2 note and indicate that it's a cell number
- If they offer a relatives or friends number do not enter the number. Explain that we can only use numbers that they ""own"" for this section

/"Contact Telephone"

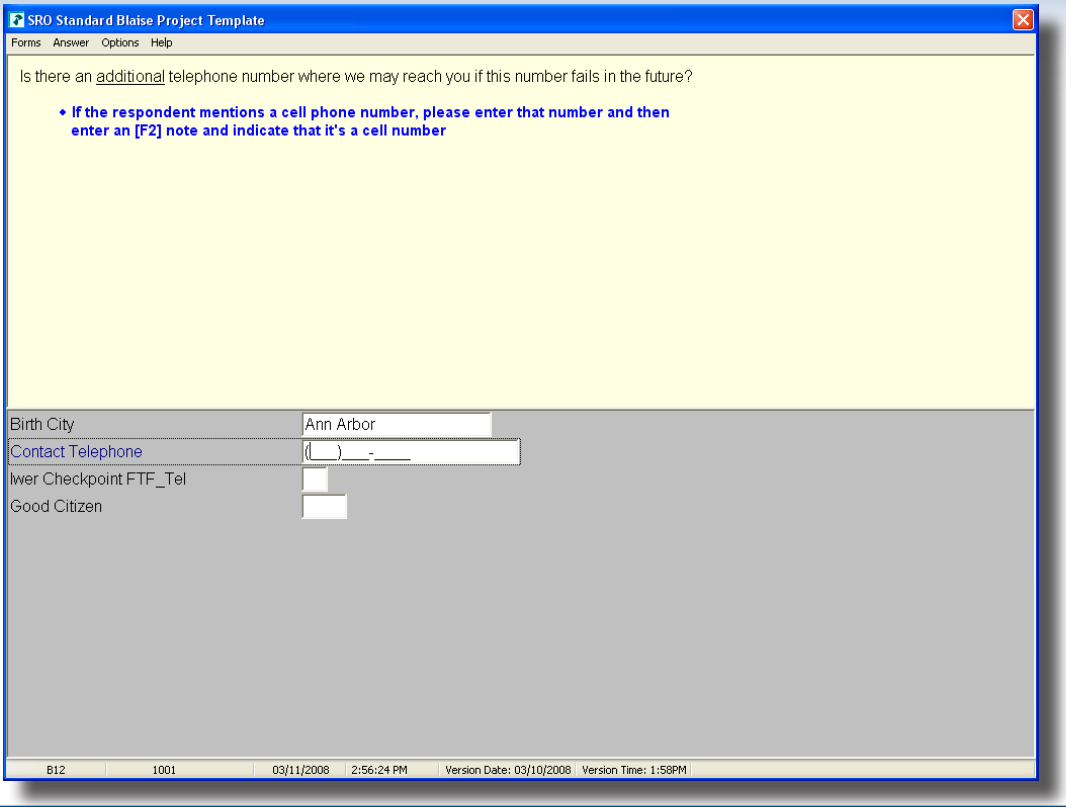
String; Telephone Mask, width = 10

Specification

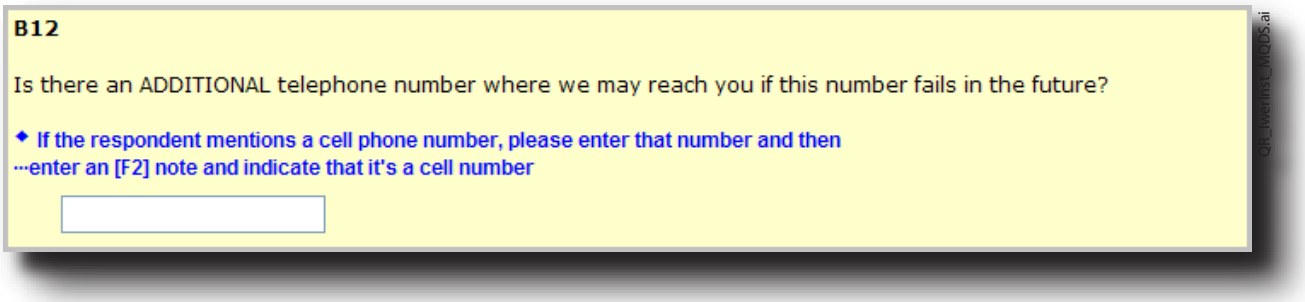
```
B12     (B12)
        "Is there an @Uadditional@U telephone number where we may
        reach you if this number fails in the future?

        @/@/@|@Dw@D @IIf the respondent mentions a cell phone number
        please enter that number then@/@|···enter an F2 note and in-
        dicate that it's a cell number@I

        @/@/@|@Dw@D @IIf they offer a relatives or friends number do
        not enter the number.··Explain that@/@|···we can only use
        numbers that they ""own"" for this section." /
        "Contact Telephone" :

        TPhone
```

Programming

Blaise DEP



MQDS



Context Related Instructions

B20

[F1] - HELP
HH Member #1

What is your date of birth?

- ENTER MM/DD/YYYY, with or without slashes

/"Respondent DOB"

DateType1; Date Mask; Range for Year 1986-2007

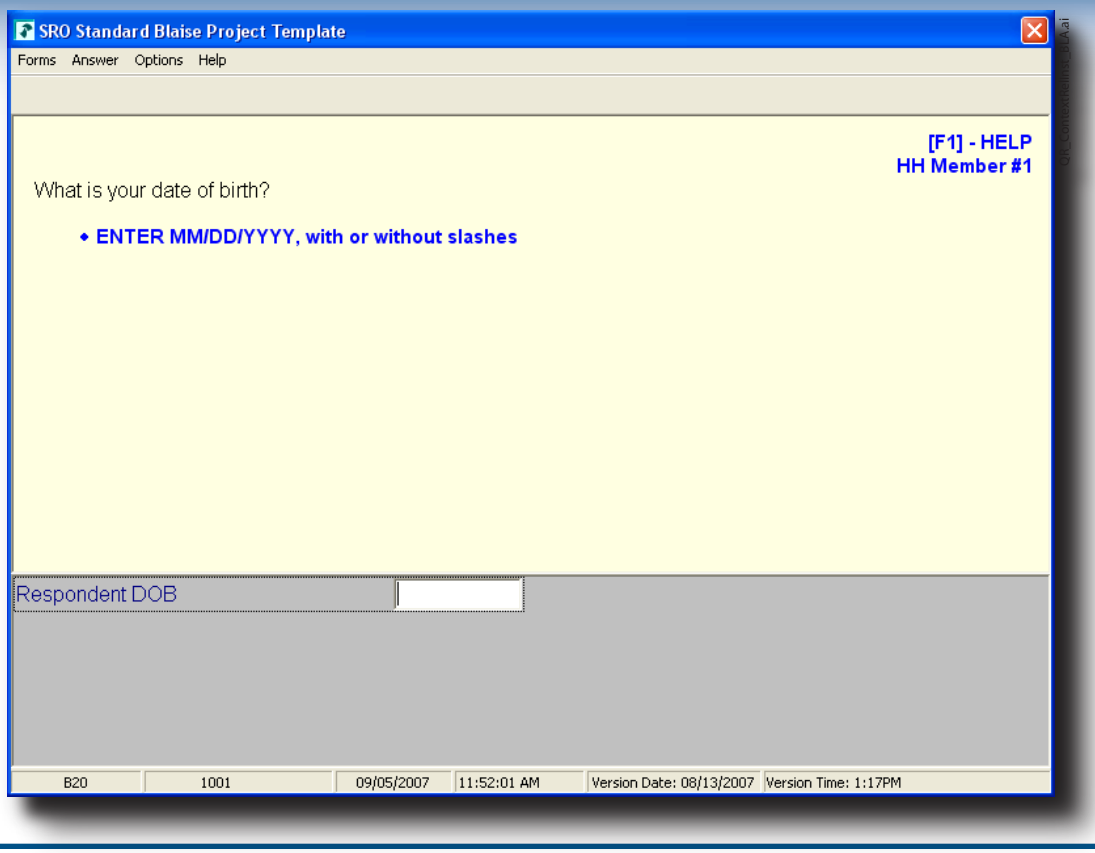
Hard Consistency Check: Check B20 Year of Birth must be later than 1985;
"The birth year must be later than 1985"

Specification

```
B20     (B20)
        "@>@I[F1] - HELP@/HH Member #1@I@<
        What is your date of birth?

        @/@/@|@Dw@D @IENTER MM/DD/YYYY, with or without slashes@I" /
        "Respondent DOB" :

        TDate
```

Programming

Blaise DEP



MQDS

Date Type

B20

[F1] - HELP
HH Member #1

What is your date of birth?

- ENTER MM/DD/YYYY, with or without slashes

/"Respondent DOB"

DateType1; Date Mask; Range for Year 1986-2007

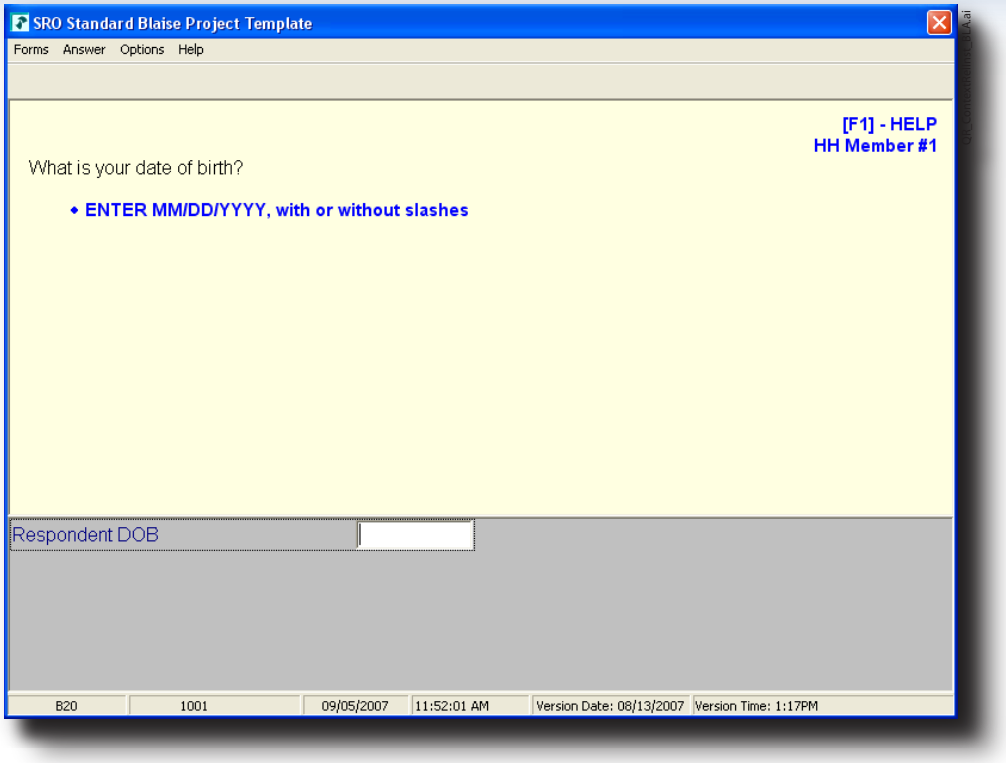
Hard Consistency Check: Check B20 Year of Birth must be later than 1985;
"The birth year must be later than 1985"

Specification

```
B20     (B20)
        "@>@I[F1] - HELP@/HH Member #1@I@<
        What is your date of birth?

        @/@/@|@Dw@D @IENTER MM/DD/YYYY, with or without slashes@I" /
        "Respondent DOB" :

        TDate
```

Programming

Blaise DEP



MQDS

Date Type (Alternate Version)

Specification

B20Month

[F1] - HELP
HH Member #1

What is your date of birth?

MM/DD/YYYY

♦ ENTER month

/"Respondent DOB - Month"

Jan	1
Feb	2
March	3
April	4
May	5
June	6
July	7
Aug	8
Sept	9
Oct	10
Nov	11
Dec	12

B20Day

[F1] - HELP
HH Member #1

(What is your date of birth?)

MM/**DD**/YYYY

♦ ENTER day

/"Respondent DOB - Day"

Integer; range 1-31

Hard Consistency Check: Check B20 Day If month = Feb and day > 28, then check for Leap Year

B20Year

[F1] - HELP
HH Member #1

(What is your date of birth?)

MM/DD/**YYYY**

♦ ENTER year

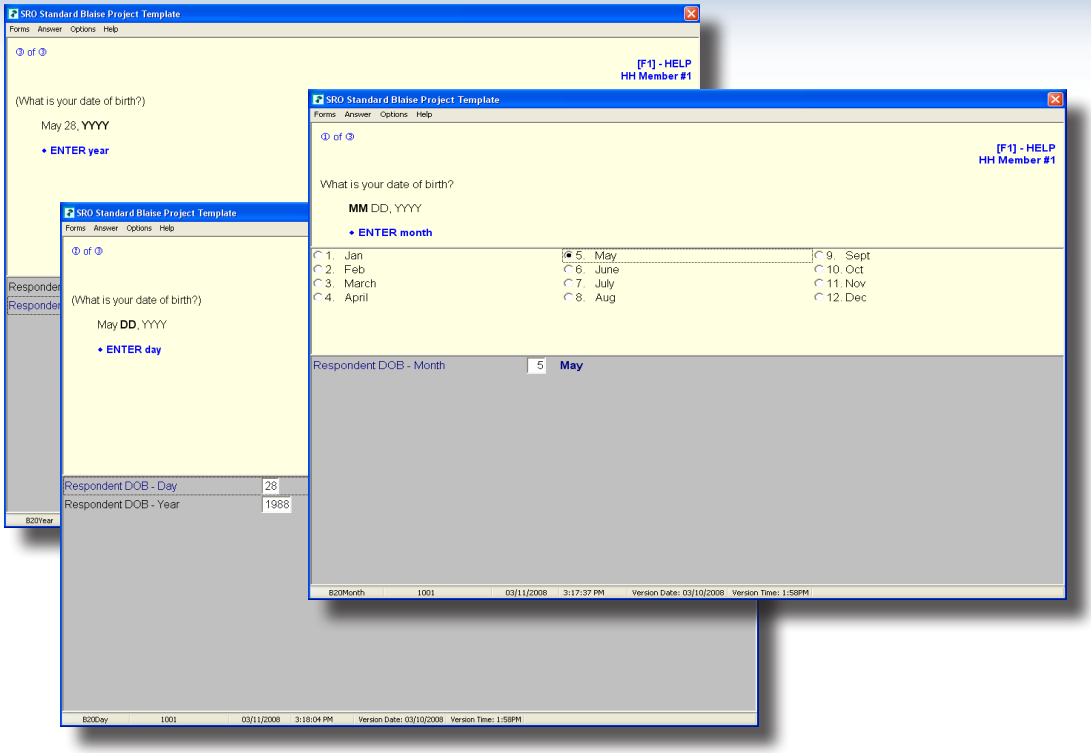
/"Respondent DOB - Year"

Integer; range 1986-2007

Programming

```
B20Month        (B20Month)
                "@Ej@E @Wof@W @El@E @>@I[F1] - HELP@/HH Member #1@I@<
                What is your date of birth?

                @/@/@|@BMonth@B DD, YYYY

                @/@/@|@Dw@D @IENTER month@I" /
                "Respondent DOB - Month" :

                TMonth

B20Day          (B20Day)
                "@Ek@E @Wof@W @El@E @>@I[F1] - HELP@/HH Member #1@I@<
                (What is your date of birth?)

                @/@/@|^B20Month @BDD@B, YYYY

                @/@/@|@Dw@D @IENTER day@I" /
                "Respondent DOB - Day" :

                TDay1_31            {1..31}

B20Year         (B20Year)
                "@El@E @Wof@W @El@E @>@I[F1] - HELP@/HH Member #1@I@<
                (What is your date of birth?)

                @/@/@|^B20Month ^B20Day, @BYYYY@B
                @/@/@|@Dw@D @IENTER year@I" /
                "Respondent DOB - Year" :

                TYear00_2007            {1900..2007}
```

Blaise DEP



MQDS

Array Series

Specification

B23
How many children do you have?

/"Number of Children"

Integer; Range 0 - 10

Programmer Note: If number of children is greater than zero, then ask the following questions for each child

B24Name
What is [the/the oldest/the next] child's first name?

/"Child First Name"

String; width = 10

Fill Logic:
If only 1 child, then Fill = the
If more than 1 child and the first time through, then Fill = the oldest
If more than 1 child and not the first time through, then Fill = the next

B24Age
How old is [Child's Name]?

/"Child Age"

Integer; range: 1-20

Fill Logic:
Use name from B24Name

Soft Consistency Check: Child's age must be less than R's age - 15;
"Age is within 15 years of R's age (Display R's Age)
Does this mean you were ^xRAgeMinus15 years old when this child was born?
♦ IF Yes, SUPPRESS Signal"

B24Gender
♦ IF necessary:

Is [Child Name] a male or female?

/"Child Gender"

Male	1
Female	2

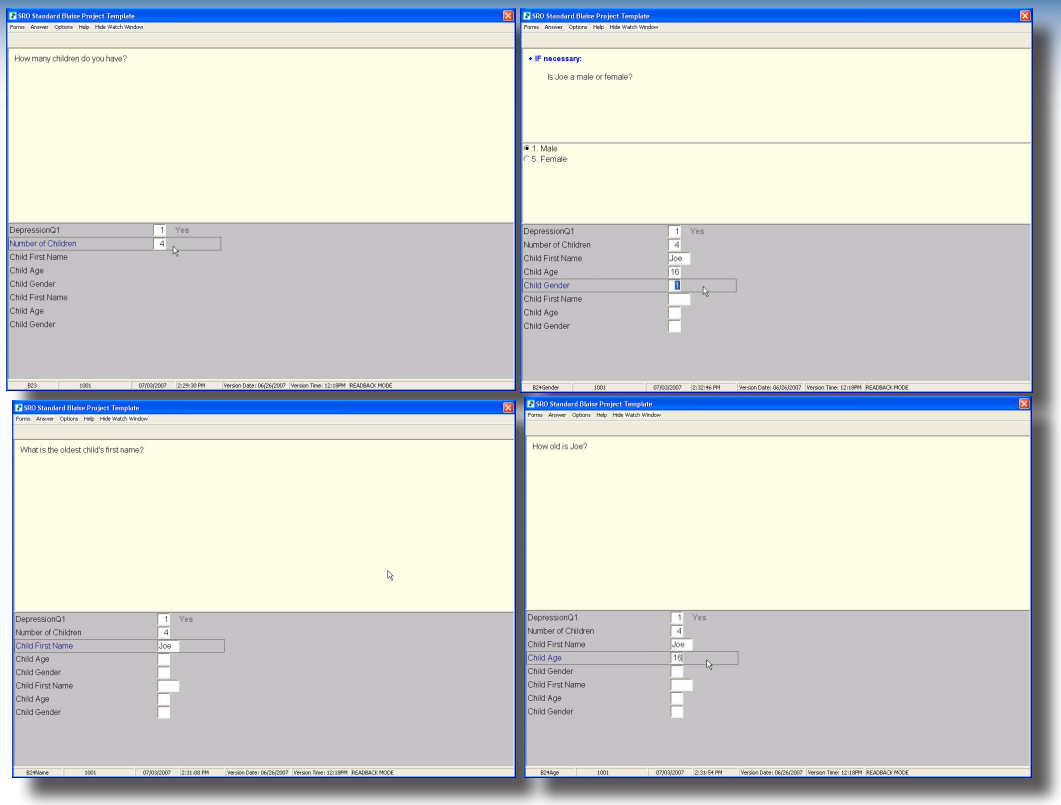
Fill Logic:
Use name from B24Name

Programming

```
B23     (B23)
        "How many children do you have?" /
        "Number of Children":

        0..10

    BLOCK   BChildInfo
        PARAMETERS
            LineNum : INTEGER
        AUXFIELDS
            xoldest_next    : STRING[15]
        FIELDS
             Name   (B24Name)
                    "What is ^xoldest_next child's first
                     name?" /
                    "Child First Name" :

                    STRING[10]

             Age    (B24Age)
                    "How old is ^Name?" /
                    "Child Age" :

                    TAge

             Gender   (B24Gender)
                    "@Dw@D @IIF necessary:@I

                    @/@/@|Is ^Name a male or female?" /
                    "Child Gender" :

                    TGender
```

```
        RULES
            IF B23 = 1 THEN
                xoldest_next := 'the'
            ELSEIF B23 > 1 THEN
                IF LineNum = 1 THEN
                    xoldest_next := 'the oldest'
                ELSE
                    xoldest_next := 'the next'
                ENDIF
            ENDIF
            Name
            Age
            Gender
        ENDBLOCK    {BChildInfo}
    FIELDS
        B24:Array [1..10] of BChildInfo
```

Blaise DEP



MQDS

B23

How many children do you have?

GOTO B25

Name

What is [the / the oldest / the next] child's first name?

Age

How old is [NAME]?

GOTO Gender

SOFT Edit

Valid condition: (Age <= (xRAge - 15)) INVOLVING (Age)

Error returned to the user: ◆ Age is within 15 years of R's age ([AGE B20])

Does this mean you were [xRAge - Age] years old when this child was born?

◆ IF Yes, SUPPRESS Signal

Involved fields: Section_B.B24.Age

Gender

◆ IF necessary:

Is [NAME] a male or female?

1 Male

5 Female

QR_Array_MQDS.ai

Checkpoints Programmer Explicit Checkpoint

B20ChkPt (See B20Mont, B20Day, and B20Year)
Data checkpoint for completeness of Respondent DOB

/"Data Checkpoint: Respondent DOB"

DOB Complete	1
DOB Incomplete	5

Explicit

B20Month = Response AND B20Day = Response AND B20Year = Response = DOB Complete

Specification

```
B20ChkPt    (B20ChkPt)
        "Data Checkpoint for Respondent DOB completeness" /
        "Data Checkpoint: Respondent DOB" :

        TDOBChkPt


RULES

        B20ChkPt.KEEP
        IF B20Month = RESPONSE AND B20Day = RESPONSE AND B20Year = RE-
        SPONSE THEN
            B20ChkPt := DOBComplete
        ELSE
            B20ChkPt := DOBInComplete
        ENDIF
```

Programming

Blaise DEP

No Image

MQDS

No Image

Appendix B • Glossary of Terms

CAI

Computer-Assisted Interviewing.

Check

Consistency check that will not allow Interviewer to continue until the inconsistency has been resolved in the data; also known as a "Hard Check".

Consistency Check

Available in two types; "Signal" or "Soft Check" allows the Interviewer to suppress the check and continue and the "Check" or "Hard Check" which forces the Interviewer to resolve the inconsistency.

DateTypes

A date value that holds the basic information for a date, allowing the storage and retrieval of dates.

Edit Mask

A manner in which data entry of currency amounts, telephone numbers, etc are displayed to the Interviewer during entry.

Enumerated

A type defined as a discrete range in which each category is specified as an identifier; also known as a "coded" question.

Explicit Checkpoint

Checkpoint is an explicit field in the data file but it is programmed (not interviewer entered).

Field Description

A second or alternative text that describes the field and is used in the Field Pane; it also can be used in SAS output.

Field Name

Primary identifier of a field; e.g. HeatHome, RGender, etc.

Field Tag

A secondary identifier of a field; e.g. A1, A2, A3, etc.

Implicit Checkpoint

Checkpoint is implicit and a guide for logic only. The field will not appear in the data file.

Looped Series (Arrays)

A series of questions repeated multiple times. The data is arrayed.

On Screen Iwer Checkpoint

Checkpoint that is on screen and requires an interviewer response.

Open-Ended

A free-form text variable that can contain text of varying length.

Preload

This is previously known data that is loaded into a case when the case is started.

QxQ's

Aides that assist the Interviewer by providing contextual information or definitions of certain terms in the question text.

Range

Sets the upper and lower boundaries for integer and real number fields.

Set Type

This is a multiple response type; also known as "Multiple Choice".

Signal

Consistency check that will allow Interviewer to continue and override the inconsistency in the data; also known as a "Soft Check".

Appendix C • HHL Code

Introduction

Following is an example of programming code for an HHL roster and selection of a Respondent 18 or older.

```
FIELDS
A4     (A4)
        "@BA4.@B
        @/@/Now, in order to select the right person to speak with, I need you to think of every-
        one who was living in your household on August 15, 2005 before Hurricane Katrina.··I am
        going to ask you to list for me everyone who was living in your household before Hurricane
        Katrina.··I am interested in those people who were living in your household @Bthen@Bwho
        are at least 18 years old @Bnow@B.··Let's start with you.

        @/@/(What is your sex?)" /
        "Sex Informant" :

        TGender

A5(A5)
        "@BA5.@B
        @/@/What is your current age?" /
        "Age Informant" :

        TAge18_120

A6     (A6)
        "@BA6.@B
        @/@/Was there anyone else living in your household on August 15, 2005 before Hurricane Ka-
        trina who is now at least 18 years old?" /
        "Other HH Members" :

        TYesNo

BLOCK BA7
    PARAMETERS
     LineNum : INTEGER
    AUXFIELDS
     xthat_thenext   : STRING[10]
    FIELDS
     PersonNum   / "Person Number" : 1..15

     A7a  (A7a)
       "@BA7a.@B
       @/@/What is ^xthat_thenext person's sex?" /
       "Sex HH Member" :

       TGender

     A7b  (A7b)
       "@BA7b.@B
       @/@/What is ^xthat_thenext person's age?" /
       "Age HH Member" :

       TAge
     A7c  (A7c)
       "@BA7c.@B
       @/@/What is this person's relationship to you?" /
       "Rel HH Member" :

       TRelation

     A7cLabel (A7cLabel) / "Rel Label" :

       STRING[25]

     A7d  (A7d)
       "@BA7d.@B
       @/@/@Dw@D @IPlease specify@I" /
       "Other Rel HH Member" :

       STRING[100]

     A7e  (A7e)
```

```
    "@BA7e.@B
    @/@/Is this person still currently living in your household?" /
    "Still in HH" :

    TYesNo

  A7f   (A7f)
    "@BA7f.@B
    @/@/Was there anyone else living in your household before Hurricane
    Katrina who is now at least 18 years old?" /
    "Other HH Members" :

    TYesNo

RULES
  PersonNum.KEEP
  A7a.KEEP
  A7b.KEEP
  A7c.KEEP
  A7cLabel.KEEP
  A7d.KEEP
  A7e.KEEP
  A7f.KEEP
  IF LineNum > 1 THEN
    IF A7a <> EMPTY THEN
    PersonNum := LineNum
    ENDIF
  ENDIF
  IF LineNum = 2 THEN
    xthat_thenext := 'that'
  ELSEIF LineNum > 2 THEN
    xthat_thenext := 'the next'
  ENDIF
  IF LineNum = 1 THEN
    PersonNum.SHOW
    A7a.SHOW
    A7b.SHOW
    A7c.SHOW
    A7cLabel.KEEP
    A7cLabel := 'Informant'
    A7cLabel.SHOW
    A7d.KEEP
    A7e.KEEP
    A7f.KEEP
  ELSE
    PersonNum.SHOW
    A7a
    A7b
    A7c
    IF A7c = RESPONSE THEN
      A7cLabel.KEEP
      A7cLabel := ''
      IF A7c.ORD = 0 THEN
        A7cLabel := 'Informant'
      ELSEIF A7c.ORD = 1 THEN
        A7cLabel := 'Wife/Female Partner'
      ELSEIF A7c.ORD = 2 THEN
        A7cLabel := 'Husband/Male Partner'
      ELSEIF A7c.ORD = 3 THEN
        A7cLabel := 'Mother'
      ELSEIF A7c.ORD = 4 THEN
        A7cLabel := 'Father'
      ELSEIF A7c.ORD = 5 THEN
        A7cLabel := 'Sister'
      ELSEIF A7c.ORD = 6 THEN
        A7cLabel := 'Brother'
      ELSEIF A7c.ORD = 7 THEN
        A7cLabel := 'Daughter'
      ELSEIF A7c.ORD = 8 THEN
        A7cLabel := 'Son'
      ELSEIF A7c.ORD = 9 THEN
        A7cLabel := 'Grandmmother'
      ELSEIF A7c.ORD = 10 THEN
        A7cLabel := 'Grandfather'
      ELSEIF A7c.ORD = 11 THEN
        A7cLabel := 'Roommate'
      ELSEIF A7c.ORD = 12 THEN
        A7cLabel := 'Aunt'
      ELSEIF A7c.ORD = 13 THEN
        A7cLabel := 'Uncle'
```

```
          ELSEIF A7c.ORD = 14 THEN
            A7cLabel := 'Niece'
          ELSEIF A7c.ORD = 15 THEN
            A7cLabel := 'Nephew'
          ELSEIF A7c.ORD = 16 THEN
            A7cLabel := 'Cousin'
          ELSEIF A7c.ORD = 17 THEN
            A7cLabel := 'Granddaughter'
          ELSEIF A7c.ORD = 18 THEN
            A7cLabel := 'Grandson'
          ELSEIF A7c.ORD = 97 THEN
            A7cLabel := 'Other'
          ENDIF
          A7cLabel.SHOW
          IF A7a = MALE THEN
            CHECK
              A7c.ORD = 2 OR A7c.ORD = 4 OR A7c.ORD = 6 OR A7c.ORD = 8 OR
              A7c.ORD = 10 OR A7c.ORD = 11 OR A7c.ORD = 13 OR A7c.ORD = 15 OR
              A7c.ORD = 16 OR A7c.ORD = 18 OR A7c.ORD = 97 OR
              A7a = NONRESPONSE OR A7c = NONRESPONSE
              "@D@Rw@R@D @R@BThe Relationship code should match the
              gender@B@R"
          ELSEIF A7a = FEMALE THEN
            CHECK
              A7c.ORD = 1 OR A7c.ORD = 3 OR A7c.ORD = 5 OR A7c.ORD = 7 OR
              A7c.ORD = 9 OR A7c.ORD = 11 OR A7c.ORD = 12 OR A7c.ORD = 14 OR
              A7c.ORD = 16 OR A7c.ORD = 17 OR A7c.ORD = 97 OR
              A7a = NONRESPONSE OR A7c = NONRESPONSE
              "@D@Rw@R@D @R@BThe Relationship code should match the
              gender@B@R"
          ENDIF
        ENDIF   {A7c = RESPONSE}
        IF A7c = Other THEN
          A7d
        ENDIF
        A7e
        A7f
      ENDIF
    ENDBLOCK{BA7}

    TABLE   THHL
      LOCALS
        I   : INTEGER
      FIELDS
        HHL / "HH Listing" : ARRAY [1..15] OF BA7
      RULES
        HHL[1].KEEP (1)
        HHL[1].SHOW (1)
        FOR I := 2 TO 15 DO
          IF HHL[I - 1].A7f = YES THEN
            HHL[I] (I)
          ENDIF
        ENDDO
      LAYOUT
        BEFORE HHL[1] GRID Table2
    ENDTABLE{THHL}
FIELDS
    HHList  : THHL

    HHLLockFlag / "Flag to Lock HHL" : TYesNo

    A8(A8)
      "@BA8.@B
      @/@/You have said the people living in your household before Hurricane Katrina who are now at least
      18 years old are:

      @/@|@|@B@L^xHHListing@L@B

      @/@/Does that include everyone who is now at least 18 years old and who was living in your house-
      hold on August 15, 2005 before Hurricane Katrina?

      @/@/@|@Dw@D @IIf ""No"", PRESS the [UP] arrow to return to the roster
      screen and correct the roster or add to@/@|···it as needed@I

      @/@/@|@Dw@D @IOtherwise, enter ""1"" and press [Enter] to continue.··Once you do this, you @Uwill
      not@U be allowed to@/@|···return to the roster to edit any information, so confirm all information
      before filling in this field with@/@|···any answer" /
      "HHL Complete & Confirmed" :

      TYesRoster
```

```
HH18Count   (HH18Count)
    "Count of HH Members 18yrs+ Currently in HH" /
    "Count of HH Members 18yrs+ Currently in HH" :

    0..15

HH18PersNum        (HH18PersNum)
       "Person Number of HH Members 18yrs+ Currently in HH" /
       "Person Number of HH Members 18yrs+ Currently in HH" :

       ARRAY[1..15] OF 1..15

RanSelFlag   (RanSelFlag)
    "Random Selection Flag to Prevent Re-Randomization" /
    "Random Selection Flag" :

    TYesNo

RanNumSel   (RanNumSel)
    "Random Number Selected from HH18Count" /
    "Random Number Selected from HH18Count" :

    1..15

BLOCK BRespInfo "Selected Respondent Info"
  FIELDS
    RespPNum       (RespPNum)
      "Respondent Person Number" /
      "Respondent Person Number" :

      1..15

    RespGender     (RespGender)
      "Respondent Gender" /
      "Respondent Gender" :

      TGender

    RespAge        (RespAge)
      "Respondent Age" /
      "Respondent Age" :

      TAge

    RespRel        (RespRel)
      "Respondent Relation" /
      "Respondent Relation" :

      TRelation

    RespRelLabel(RespRelLabel)
      "Respondent Relation Label" /
      "Respondent Relation Label" :

      STRING[25]
  ENDBLOCK{BRespInfo}
FIELDS
  RespInfo: BRespInfo

  RInform (RInform)
      "Is Informant Selected Respondent?" /
      "Is Informant Selected Respondent" :

      TYesNo

RULES
  A4
  A5
  A6
  HHList.KEEP
  HHList.HHL[1].PersonNum := 1
  HHList.HHL[1].A7a := A4
  HHList.HHL[1].A7b := A5
  HHList.HHL[1].A7c := Informant
  HHList.HHL[1].A7f := A6
  xHHListing.KEEP
  xHHListing := EMPTY
  FOR I := 1 TO 15 DO
    IF HHList.HHL[I].A7c <> Other THEN
```

```
    xRelationship := HHList.HHL[I].A7cLabel
   ELSE
    xRelationship := HHList.HHL[I].A7d
   ENDIF
   IF HHList.HHL[I].PersonNum <> EMPTY AND
    HHList.HHL[I].A7b > 17 AND xHHListing = EMPTY THEN
    xHHListing := xRelationship+', Age: '+STR(HHList.HHL[I].A7b)

    J := I + 1
   ENDIF
 ENDDO
 FOR I := J TO 15 DO
   IF HHList.HHL[I].A7c <> Other THEN
    xRelationship := HHList.HHL[I].A7cLabel
   ELSE
    xRelationship := HHList.HHL[I].A7d
   ENDIF
   IF HHList.HHL[I].PersonNum <> EMPTY AND HHList.HHL[I].A7b > 17 THEN
    xHHListing := xHHListing+'@/@|@|'+xRelationship+', Age: '+STR(HHList.HHL[I].A7b)
   ENDIF
 ENDDO
 HHLLockFlag.KEEP
 IF A6 = YES AND HHLLockFlag = EMPTY THEN
   HHList
 ELSEIF HHLLockFlag = YES THEN
   HHList.KEEP
 ENDIF   {A6 = YES}
 A8
 RanSelFlag.KEEP
 RanNumSel.KEEP
 RespInfo.KEEP
 RInform.KEEP
 IF A8 = YES THEN
   HHLLockFlag := YES
   HH18Count := 0
   FOR I := 1 TO 15 DO
    IF SampleType = NonRDD THEN
      IF HHList.HHL[I].PersonNum <> EMPTY AND
       HHList.HHL[I].A7b > 17 THEN
       HH18Count := HH18Count + 1
       HH18PersNum[HH18Count] := I
      ENDIF
    ELSEIF SampleType = RDD THEN
      IF HHList.HHL[I].PersonNum = 1 OR
       (HHList.HHL[I].PersonNum <> EMPTY AND
       HHList.HHL[I].A7b > 17 AND
       HHList.HHL[I].A7e = YES) THEN
       HH18Count := HH18Count + 1
       HH18PersNum[HH18Count] := I
      ENDIF
    ENDIF
   ENDDO
   IF HH18Count = 1 THEN
    RespInfo.RespPNum   := HHList.HHL[1].PersonNum
    RespInfo.RespGender := HHList.HHL[1].A7a
    RespInfo.RespAge      := HHList.HHL[1].A7b
    RespInfo.RespRel:= HHList.HHL[1].A7c
    RespInfo.RespRelLabel   := HHList.HHL[1].A7cLabel
   ELSE
    IF RanSelFlag = EMPTY THEN
     RanNumSel := RANDOM (HH18Count) + 1
     RanSelFlag := YES
    ENDIF
    RespInfo.RespPNum   := HHList.HHL[HH18PersNum[RanNumSel]].PersonNum
    RespInfo.RespGender := HHList.HHL[HH18PersNum[RanNumSel]].A7a
    RespInfo.RespAge:= HHList.HHL[HH18PersNum[RanNumSel]].A7b
    RespInfo.RespRel:= HHList.HHL[HH18PersNum[RanNumSel]].A7c
    RespInfo.RespRelLabel   := HHList.HHL[HH18PersNum[RanNumSel]].A7cLabel
   ENDIF
   IF RespInfo.RespRel = Informant THEN
    RInform := YES
   ELSE
    RInform := NO
   ENDIF
```

Index