

The American Time Use Survey Data Collection Instrument

*David Altvater, William Dyer, Christian Borillo, Roberto Picha
United States Bureau of the Census*

This paper reports the results of work undertaken by the United States Census Bureau staff. It has undergone a US Census Bureau review more limited in scope than that given to official Census Bureau publications. This report is released to inform interested parties and to encourage discussion of work in progress.

1. Introduction

The American Time Use Survey (ATUS) Blaise data collection instrument, sponsored by the U.S. Bureau of Labor Statistics, developed by the U.S. Bureau of the Census, and conducted in Computer Assisted Telephone Interviewing (CATI) mode, is designed to provide nationally representative estimates of time that Americans spend in various activities. The data are used to measure the value of unpaid, productive work such as housework and childcare, and to measure nonproductive activities such as waiting in line and commuting to work.

The ATUS instrument is particularly challenging because of the dynamic reconciliation of complex time relationships represented in the main module of the instrument, the Diary section. In addition, another section of the instrument is dependent on the time data in the Diary and must be updated from that data.

The ATUS Diary is the section of the instrument that is used to collect data about various activities performed throughout the day. People tend to go through a routine but may not remember how long an activity took (duration entries) or exactly what time they ended the activity, unless it is a regularly scheduled event such as leaving work or watching a television show. A typical interview would collect most of a person's activities and times during the day and then insert, delete, or modify entries as the respondent recalled events and times during the course of the interview.

In the Diary section of the ATUS instrument, time data are collected for activities that cover a 24-hour time period, as reported by the respondent. These time data are collected either by duration units (hours and minutes) or by wall clock stop time of the activity. As the data are collected in the Diary module, the time relationships within and across rows are constantly recalculated relative to the time entries in the previous row. Within a row, if duration is entered, stop time is calculated, and vice versa. The calculation is based on the stop time of the previous row.

Figure A – Activities and Time Relationships in a Partially Populated ATUS Diary Section Grid.

ID	Start	Activity	TIME	Hrs	Mins	Stop	Who	Who_2	Where	Where specify
[1]	4:00AM	Sleeping	1	2		6:00AM				
[2]	6:00AM	Grooming	2	0	22	6:22AM				
[3]	6:22AM	Watching TV/grooming	1		15	6:37AM	0		1	Respondent's
[4]	6:37AM	Preparing meals and snacks	1		5	6:42AM	0		1	Respondent's
[5]	6:42AM	Commute to work	1		45	7:27AM	55		12	Car, truck, or
[6]	7:27AM	Working at main job	2	4	3	11:30AM			2	Respondent's
[7]	11:30AM	Eating and drinking/working on Blaise	1		30	12:00PM	55		2	

00000001 WHERE 9:09:49 AM 2/14/2003

Backward movement and changes are possible, as are row insertions and row deletions. Particularly challenging to maintaining the accuracy of the data relationships is the need to track several layers of time data, thereby creating a virtual three-dimensional table (row, column, time [previous and current]). The two layers of time are the “current” data displayed in the Diary (layer 1), and the “previous” responses, data which are not displayed but are retained in “shadow” fields (layer 2).

This paper summarizes the two approaches that were attempted in the development of the ATUS Diary modules and presents some of the lessons learned during the process.

The first approach allowed the interviewers to key data in either the duration fields (hours and minutes) or the stop time field, without specifying which, and the program determined which method was used. This non-specified, or implied, data entry method was not used for production because of programming complexities it introduced.

The second approach, which is currently used for production ATUS interviewing, also allows the interviewers to enter the data using either duration or stop time, but the interviewers must first select the method they will use on any given row. Then, when data are keyed into either the duration or stop time fields, the program uses the selected method of time data entry to determine which way the time calculations are handled by Blaise. At any time, the interviewer can switch methods by toggling the value of a row-level field in the Diary table. This paper will discuss the pros and cons of these two approaches and the avenue used for development of each, including some lessons learned.

2. Version 1 – Non-Specified Method of Time Entry

The first attempt for the Diary section was developed as a working demonstration prototype. It allowed the interviewer to enter time data either as duration or as stop time without requiring the interviewer to specify which method was used. The code was written to allow free navigation, insertion, deletion, and sandwich insertions (splitting an activity). The code reconciled all of the time entries across the rows of events. This presented a problem because the program first needed to determine which entry method was used. Based on the method used by the interviewer, the program then used the appropriate section of code to perform calculations on subsequent time entries past the current Diary section row.

This prototype was developed to demonstrate the kinds of movement, time entry method flexibility, and accounting that might be possible for a Diary.

When entries in the Diary were added from start to finish, everything worked as expected for time calculations. But when the grid was already populated and changes were made (adding, deleting, inserting rows or modifying the content of rows), the time calculations had to be recomputed. Reconciling the two types of entries, stop time and duration, was difficult because the program needed to remember which method of time calculation was used on any given row. This information was not tracked in early Version 1 Diary code. Techniques to record the nature of the entries were introduced but quickly became unmanageable because of the need to retain previously entered data that is not included in the Diary display (since it shows only current data entries). Data fields which are not displayed but which are retained to help determine the nature of respondent activity and previous responses are called shadow variables.

When a new row was inserted, hour, minute and/or stop time entries were empty/missing – only the start time was present since it was populated with the previous stop time entry. Users were able to update any field freely. This set up two scenarios: 1) the hour-minute entry may update the stop time field or 2) the stop time entry may update the hour and minute fields. The following examples and pseudo code show the process of making these calculations:

If the time duration was entered, the instrument would calculate the stop time for the present row and adjust entries in the subsequent rows of the Diary.

If the stop time was entered, then the duration time in hours and minutes was calculated, adjusted and propagated to the following rows in the grid.

```
If hourduration = response or minutesduration = response then
    stop = (start + hourduration + minutesduration)
else
    timeresult = stop-start
    hourduration = timeresult.hour
    minutesduration = timeresult.minutes
endif
```

Here is an example to illustrate how the calculations became a challenge for the Diary programmer.

Entry					Calculated				
Row	Start	Hour	Minute	Stop	Row	Start	Hour	Minute	Stop
1) Sleep	4:00	2		6:00 AM	1) Sleep	4:00	2	0	6:00 AM
2) Eat			30	6:30 AM	2) Eat	6:00	0	30	6:30 AM
3) Work		9		3:30 PM	3) Work	6:30	9	0	3:30 PM

Reported time entries are in gray.
Rules execution recalculates all of the time entries.

In this example, each stop time entry updates the hours and minutes and the next row's start time. If the time duration was entered, the instrument calculates the stop time for the present row and adjusts entries in subsequent rows. Notice that once the fields are calculated we cannot easily identify the method used to derive the entries. This is due to the fact that using the “response” query only works the first time through the Diary as the interviewer enters times for the first time. The second and later time(s) cannot rely on the “response” test to determine which method of time entry was used, because both duration and stop fields would then contain entries. In the example above, gray areas represent the data as reported by the respondent and will drive which type of time calculations are done by the program. To continue using the program to calculate the row times, shadow

variables were used to hold information about the past iteration of the rules process.

Blaise performs calculations described in the code from the top down. These “Rules” are re-executed each time a field is exited after a data change is recorded. This is the nature of interactive data collection and is what the language is designed to do, but potentially causes us some problems with the time calculations in certain circumstances.

Now, lets go back and modify the previous example by adding another activity. The respondent remembers that they spent 45 minutes commuting to work. The interviewer inserts a row for commuting, which takes place between eating breakfast and arriving at work, i.e. insert a new row - #3.

Entry					Calculated				
Row	Start	Hour	Minute	Stop	Row	Start	Hour	Minute	Stop
1) Sleep	4:00	2		6:00 AM	1) Sleep	4:00	2	0	6:00 AM
2) Eat			30	6:30 AM	2) Eat	6:00	0	30	6:30 AM
3) Commute			45	7:15 AM	3) Commute	6:30	0	45	7:15 AM
4) Work				3:30 PM	4) Work	7:15	9	0	4:15 PM

Insertion of row #3 for a fixed duration of 45 min.

May cause incorrect reporting of work stop time.

This will result in a recalculation of the stop time for entry #4, Work, ending at 4:15, when the respondent originally reported leaving work at 3:30 pm. This happens because the evaluation inside the if statement used the stop time of the row #3, Commute, as the start time of row #4, Work, and then added to row #4's start time the hours and minutes previously computed from a duration entry.

One can see that the stop time calculation is being performed when the hours and minutes are reported and each row is reconciled after any cell is exited for each entry in the table and the stop time is ignored for row #4 because it is in the “else” portion of the “if” statement. Of course we could reverse this condition and make calculations based on stop times, but we will have a similar problem reconciling the duration entries. This demonstrates the need for retaining the type of entry and being able to recalculate start times for successive rows throughout the table.

This is a simple example that illustrates a problem with adding activities. Consider what is involved in adding the overhead within the code to control deletions and sandwich insertions (splitting an event).

To control all the specific scenarios, the initial code had up to seven layers of if-else-endif statements to invoke edits. There was a great deal of logic for covering all the possible entries that could be made in the time calculation fields. This code employed numerous on/off flags, which in turn would set other shadow variables depending on conditions in the entry. The example shows that reconciling the time entries correctly in previous rows is difficult no matter which method is used to solve the problem. Accordingly, maintaining and enhancing the code became more

and more difficult as the programmer tried to implement increasingly more complex requirements and provide maximum flexibility for the interviewer.

Testing uncovered several problems with the time calculations, and the complexity of the code made it virtually impossible to fix in the time remaining to production. Therefore, a decision was made to redesign the Diary, with the goals of documenting the requirements and simplifying the code.

3. Version 2 – Specified Method of Time Entry

As explained above, The Diary section for ATUS was not working as expected. It was determined through extensive research, testing, and by evaluating problem reports, that Version 1 could not be easily patched or made production ready in the time for the ATUS Dress Rehearsal live interviewing.

The decision was made to abandon Version 1 of the Diary and build a Diary section from the beginning using a more structured design approach and the knowledge gained during our work with Version 1. This redesigned Diary became known as Version 2. The number one priority assigned to the redesigned Diary code was to keep the time calculations correct and synchronized across all rows in the Diary table.

A new Diary field, “TIME”, was created. Interviewers use it to choose which of the two methods will be used to enter row times, i.e., time duration or stop time. By default, the field is set to “1 – Time Duration” but the interviewer has the flexibility to toggle between methods at any time.

Time calculations move from the data entered forward through the table, and are based on the method selected. For example, if Time Duration is active, from the point where a time duration is entered forward, the code calculates subsequent row time values by adding the hours and minutes to the start time to populate the stop time field. If Stop Time calculations were active, then after the interviewer enters the stop time for an activity, the code calculates the hours and minutes between that row’s start and stop times. Then it populates the Hours and Minutes fields on that row, and moves forward to subsequent rows to recalculate start and stop times for them.

Also important in this method of time calculation is the use of SHOW fields to indicate, on subsequent rows, which method of time entry was used on each row. Since it could change across rows in the table, this method of graying out the “calculated time values” is visually helpful to the interviewers. The method used to enter time data on each row determines which fields are recalculated for each time event recorded in previous rows.

The code below shows how these calculations were performed.

```
{=====//rpicha-cborillo additions, 04-16-02//=====}
IF typeduration = 1 THEN
    Stop := Start + (HourDur, MinDur, 0)
ELSEIF typeduration = 2 THEN {Enter system time format }
    IF (Start = RESPONSE) AND (.Stop = RESPONSE) THEN
        IF Start <> Stop THEN
            TimeResult := Stop - Start
        ELSEIF (Start = Stop) AND PrevStopDayNumber = 1 then
            TimeResult := Stop - Start
    ENDIF
    HourDur := TimeResult.HOUR
```

```

MinDur := TimeResult.MINUTE
ENDIF
ENDIF
{=====//END_rpicha-cborillo additions, 04-16-02//=====}

```

This technique is repeated as new time data are entered. The effect on the table is unidirectional, i.e., it is processed in one direction from the first row to the last. This, plus the retained method of entry field (indicating duration or stop time), allows entries to be computed or recomputed based on new data keyed by the user, at any point in the timeline, without changing the type of calculations used in subsequent entries. The method used to compute time entries in each row must be retained regardless of insertions, deletions or modifications made in previous rows. When this method was implemented, the TIME variable (duration time calculation or stop time calculation) was displayed as a SHOW variable for the user to see, assuring the user that the reported time calculation method was retained.

In addition to adding the TIME field to Version 2 of the Diary, this version introduced another field to allow the user to specifically enter A or P for am and pm.

4. The Redesign Process

In addition to redesigning the Diary, we also put more process controls in place. With the agreement of the sponsor/client of the ATUS, the requirements were identified and documented in a systematic fashion. We identified and prioritized functions and asked our end users and the telephone center interviewers and supervisors to test the functionality of the Diary throughout its development. Their input was very important and was used as a guide for further development of a user-friendly instrument. Once the basic Diary section requirements were established, a Change Control Board was created to manage change to the requirements and to manage the addition of new features.

An important part of the redesign effort was that the specification was organized to facilitate an incremental approach to development. This insured that each layer was working as designed, and that it was ready for production and maintainable.

- Grid design was considered first. Read only fields, visibility of fields in the table, and number of rows displayed on the screen at any given time were defined and agreed upon in advance.
- The row level requirements were discussed and functional specifications were developed to support activity at the row level before adding enhancements to table levels.
- Relationship of the row or time entry to following events was addressed. The crux of the matter was to consider only time relationships within the row and not insertion activities when developing and testing this layer of the Diary.
- Perform the calculation of time entries for each row in the grid. This means that we were able to enter any number of rows in the grid and start performing modifications on any given row and allow the code to calculate and reconcile time entries in subsequent rows.
- Deletion and insertion functionality were added last.

- After the core functions were defined, implemented, and thoroughly tested, navigation requirements (including edit signals and checks) were defined more thoroughly and implemented.

The reorganized Diary reduced the number of extra variables by 50, and the number of checks and edits was reduced from fifteen to two. The redesigned Diary section of the ATUS instrument followed a structured design method. This ensured that each stage of development was tested and debugged before the next phase of development was started.

5. Lessons Learned

The first version of the Diary was developed as a prototype to demonstrate how time entries in a table can be created, modified, deleted, and kept synchronized. But, partly by design, it was not geared for production use. It was developed to demonstrate the power of Blaise to handle complex time relationships across the rows of a table and the flexibility of Blaise to allow multiple approaches to inserting, deleting, and reconciling row data in a table. It served this purpose very well even though new functionality was requested and introduced more quickly than the code could be debugged and stabilized. As a result, the code became too complex and unstable in too short a time frame to be considered production ready.

We hoped that the demonstration code and prototypes could be used for production with little modification, but the code was developed over several months and was under constant revision. While new features were being added, it was difficult to test and debug a stable set of code, and the code became very complex. Problems in one version were not solved before new functionality was introduced. Consequently, the debugging burden intensified with each new code revision. It was not likely that the Version 1 code could have been thoroughly debugged by the survey production date or that it could be easily maintained due to the fact that it was originally built to demonstrate functionality in a rapid development, single purpose mode (demo only).

We learned a lot from the demonstration Diary Version 1, however, and the research and effort it took to build it were instrumental in our ability to develop the Version 2 Diary in a relatively short period of time.

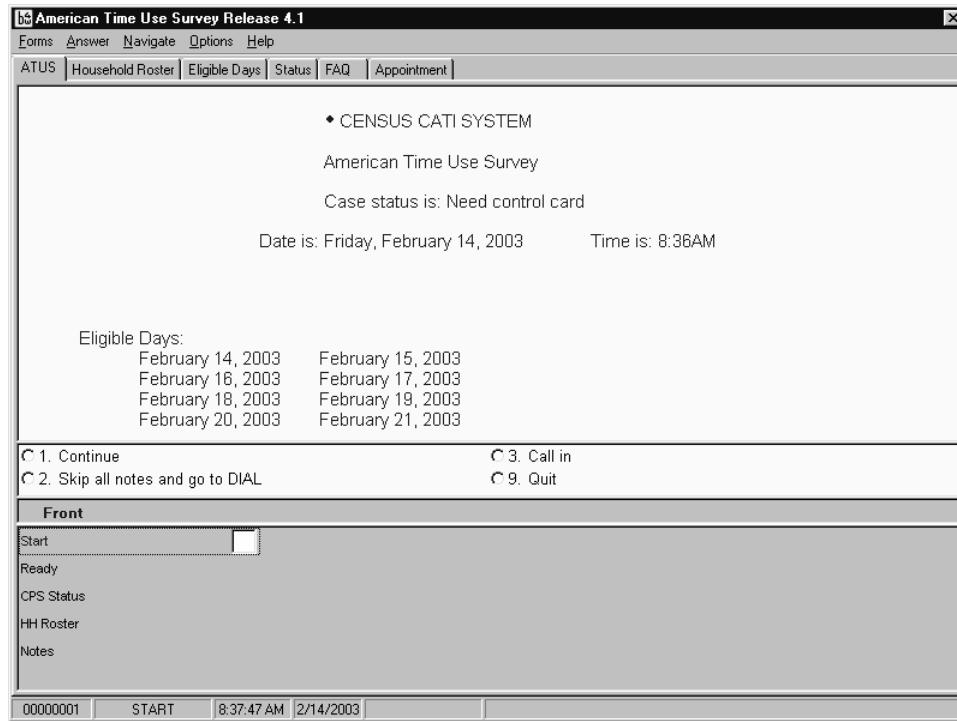
By using a more structured design approach that documented and controlled requirements and design, we were able to establish a time schedule for each release of the Version 2 Diary code. For each release, we assigned specific functionality and did not introduce new features until the “current” version was fully tested and debugged. As a result, new problems could be solved quicker and the code could be stabilized before we introduced new functionality.

This exercise underscored the need to distinguish between demonstration prototypes and production ready code. It helped us to establish a more structured design approach that we now use to repeat the successes we established during the design and development of the Version 2 Diary.

Screen captures are attached to help the reader understand the ATUS instrument and coding. Screen shots are from an example run of the Version 2 Diary section. Other examples are presented in the poster session at the International Blaise Users Conference 2003, in Copenhagen, Denmark.

The American Time Use Survey (ATUS)

Figure 1 Introduction Screen for ATUS



At the Bureau of the Census we have developed screen standards for use across both Computer Assisted Personal Interviewing (CAPI) and CATI surveys. Field representatives and telephone interviewers expect the parchment colored infopane, light grey fieldpane, function keys as described in the .bwm, parallel tabs and status bar.

Figure 2 Introduction to the Diary, Section 4 for ATUS

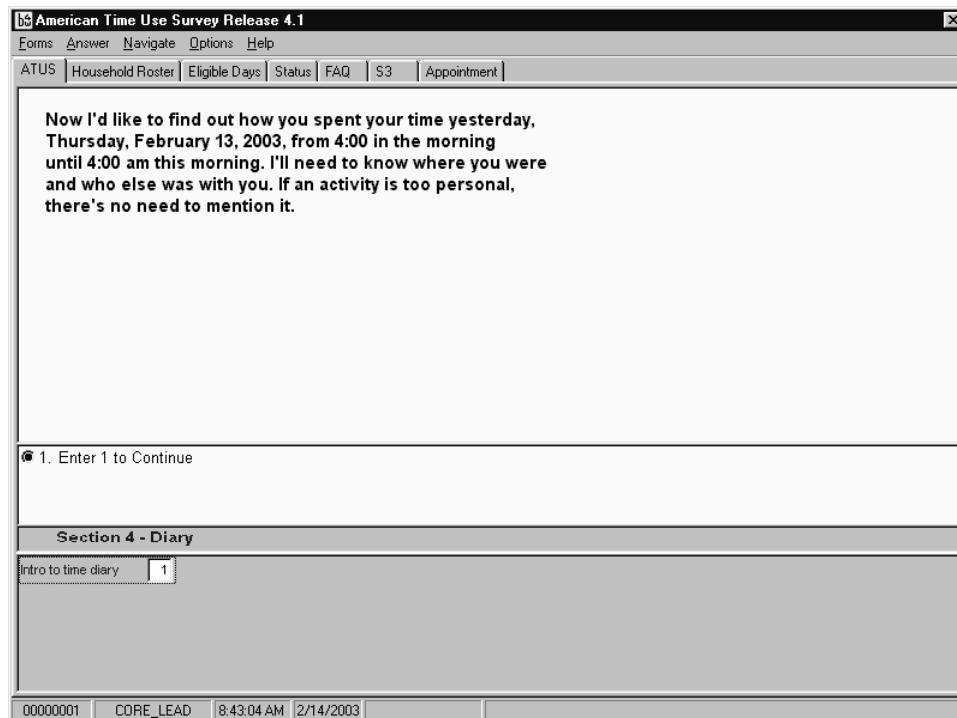


Figure 3 Introductory Diary Screen Asking about Activity.

The screenshot shows the 'American Time Use Survey Release 4.2' software interface. At the top, there's a menu bar with 'Forms', 'Answer', 'Navigate', 'Options', and 'Help'. Below the menu is a navigation bar with tabs: 'ATUS', 'Household Roster', 'Eligible Days', 'Status', 'FAQ', 'S3', 'S5', 'S8', and 'Appointment'. A sub-menu window titled 'So let's begin. Yesterday, Wednesday, at 4:00 AM, what were you doing?' is displayed. It contains instructions: 'Use the slash key (/) for recording separate/simultaneous activities.' and 'Do not use precodes for secondary activities.' followed by a list of 13 activity codes. Below this is a data entry grid for activity logs. The first row shows a log entry for 'Sleeping' starting at 4:00AM. The grid has columns for Start time, ID, Activity, TIME (with Hrs and Mins sub-columns), Stop time, Who, Who_2, Where, and Where specify. At the bottom of the grid, there are buttons for '00000001', 'ACTIVITY', 'TIME', and date/time '3/21/50 PM 3/20/2003'.

Figure 4 Time Field Entry

This screenshot shows the same software interface as Figure 3, but the sub-menu window is now asking 'How long did you spend sleeping?'. It provides instructions: 'Type 1 to enter duration (hours, minutes)' and 'Type 2 to enter stop time'. The data entry grid below is identical to Figure 3, showing the 'Sleeping' log entry. The 'TIME' column for the log entry is highlighted with a red box, indicating it is the active field for entering duration.

Time field is highlighted here, showing the default method of time data entry, "Duration".

Figure 5 More Diary Entries

Start	ID	Activity	TIME	Hrs	Mins	Stop	Who	Who_2	Where	Where specify
[1] 4:00AM		Sleeping	1	2		6:00AM				
[2] 6:00AM		Grooming	2	0	41	6:41AM				
[3] 6:41AM		Commuting	1		30	7:11AM				
[4] 7:11AM			1							
[5]										
[6]										
[7]										

Note the SHOW fields alternating as a result of toggling to stop time on row 2.

Figure 6 Row Insertion

Start	ID	Activity	TIME	Hrs	Mins	Stop	Who	Who_2	Where	Where specify
[1] 4:00AM		Sleeping	1	2		6:00AM				
[2] 6:00AM		Watching TV	1		20					
[3]		Grooming	2	6	41	6:41AM				
[4] 6:41AM		Commuting	1		30	7:11AM				
[5] 7:11AM			1							
[6]										
[7]										

Note that all other time fields are grayed out pending entry into this inserted row.

Figure 7 Sample Activities in a Partially Populated Grid for ATUS

The screenshot shows the 'American Time Use Survey Release 4.1' software window. The menu bar includes 'Forms', 'Answer', 'Navigate', 'Options', and 'Help'. The top navigation bar shows 'ATUS' and other tabs like 'Household Roster', 'Eligible Days', 'Status', 'FAQ', 'S3', 'S5', 'S8', and 'Appointment'. Below the menu is a question: 'Where were you while you were eating and drinking/working on blaise?'. It has two columns: 'PLACE' and 'MODE OF TRANSPORTATION'. A list of options follows:

C 1. Respondent's home or yard	C 12. Car, truck, or motorcycle (driver)
<input checked="" type="radio"/> 2. Respondent's workplace	C 13. Car, truck, or motorcycle (passenger)
C 3. Someone else's home	C 14. Walking
C 4. Restaurant/Bar	C 15. Bus
C 5. Place of worship	C 16. Subway/Train
C 6. Grocery store	C 17. Bicycle
C 7. Other store/Mall	C 18. Boat/Ferry
C 8. School	C 19. Taxi/Limousine Service
C 9. Outdoors away from home	C 20. Airplane
C 10. Library	C 21. Other (specify)
C 11. Other place (specify)	

Below this is a detailed activity log table:

Start	ID	Activity	TIME	Hrs	Mins	Stop	Who	Who_2	Where	Where specify
[1]		Sleeping	1:00AM	1	2	6:00AM				
[2]		Grooming	6:00AM	2	0	6:22AM				
[3]		Watching TV/grooming	6:22AM	1	15	6:37AM	0		1	Respondent's
[4]		Preparing meals and snacks	6:37AM	1	5	6:42AM	0		1	Respondent's
[5]		Commute to work	6:42AM	1	45	7:27AM	55		12	Car, truck, or i
[6]		Working at main job	7:27AM	2	4	11:30AM			2	Respondent's
[7]		Eating and drinking/working on Blaise	11:30AM	1	30	12:00PM	55		2	

At the bottom of the software window, there are status bars: '00000001 WHERE 9:09:49 AM 2/14/2003'.

Figure 8 Abbreviated Keying

Abbreviated keying for accuracy and speed

Results of previous key entry notice the first activity is automatically filled to match the pre-code 3

Figure 9 Who Code Follow Up Question

Who was in the room with you? / Who accompanied you?

On HH Roster		NonHH Family		Other NonHH	
<input checked="" type="checkbox"/> 0. Alone		<input type="checkbox"/> 51. Parents		<input type="checkbox"/> 54. Friends	
<input type="checkbox"/> 2.		<input type="checkbox"/> 52. Other non-HH family members < 18		<input type="checkbox"/> 55. Co-workers, colleagues, clients	
<input type="checkbox"/> 3.		<input type="checkbox"/> 53. Other non-HH family members 18 and older (incl. Parents-in-law)		<input type="checkbox"/> 56. Neighbors, acquaintances	
<input type="checkbox"/> 4.				<input type="checkbox"/> 57. Other non-HH children < 18	
<input type="checkbox"/> 5.				<input type="checkbox"/> 58. Other non-HH adults 18 and older	
<input type="checkbox"/> 6.					
<input type="checkbox"/> 7.					
<input type="checkbox"/> 8.					
<input type="checkbox"/> 9.					
<input type="checkbox"/> 10.					
<input type="checkbox"/> 50. All household members					

Start ID Activity TIME Hrs Mins Stop Who Who_2 Where Where specify

[1] 4:00AM		Sleeping	1	2		6:00AM			
[2] 6:00AM		Grooming	2	0	22	6:22AM			
[3] 6:22AM		Watching TV/grooming	1		15	6:37AM	0		
[4] 6:37AM			1						
[5]									
[6]									
[7]									

00000001 WHO 8:59:22 AM 2/14/2003

Notice the way the answer list is organized. The sponsor/client requested that we list different categories in columnar display based on the column header listed in blue in the infopane. This is an interesting visual way to present the user with activities for selection.

Who was with you?

Figure 10 Where Code Follow Up Question

Where were you while you were watching tv/grooming?

PLACE		MODE OF TRANSPORTATION	
<input checked="" type="radio"/> 1. Respondent's home or yard		<input type="radio"/> 12. Car, truck, or motorcycle (driver)	
<input type="radio"/> 2. Respondent's workplace		<input type="radio"/> 13. Car, truck, or motorcycle (passenger)	
<input type="radio"/> 3. Someone else's home		<input type="radio"/> 14. Walking	
<input type="radio"/> 4. Restaurant/Bar		<input type="radio"/> 15. Bus	
<input type="radio"/> 5. Place of worship		<input type="radio"/> 16. Subway/Train	
<input type="radio"/> 6. Grocery store		<input type="radio"/> 17. Bicycle	
<input type="radio"/> 7. Other store/Mall		<input type="radio"/> 18. Boat/Ferry	
<input type="radio"/> 8. School		<input type="radio"/> 19. Taxi/Limousine Service	
<input type="radio"/> 9. Outdoors away from home		<input type="radio"/> 20. Airplane	
<input type="radio"/> 10. Library		<input type="radio"/> 21. Other (specify)	
<input type="radio"/> 11. Other place (specify)			

Start ID Activity TIME Hrs Mins Stop Who Who_2 Where Where specify

[1] 4:00AM		Sleeping	1	2		6:00AM			
[2] 6:00AM		Grooming	2	0	22	6:22AM			
[3] 6:22AM		Watching TV/grooming	1		15	6:37AM	0	1	
[4] 6:37AM			1						
[5]									
[6]									
[7]									

00000001 WHERE 9:00:01 AM 2/14/2003

Where were you while you were...