System Implementation for a Blaise Multimode Web, CATI and Paper Survey

Leonard Hart, Amanda Foster-Sardenberg, Yuki Okada /Leonard Hart, Mathematica Policy Research, Inc.

This paper addresses the systems implementation underlying the use of Blaise to collect and process data in three modes: CATI, Web, and mail. Using a single Blaise instrument, data are collected or data-entered in three modes simultaneously into a common database. This approach makes it more convenient for the respondent, requires less programming, eases data management, and results in simpler and more effective survey management. However, the integration of these three disparate modes presents several challenges for underlying systems. We outline challenges and their solutions concerning the programming of the Blaise multimode instrument, the rendering of the web mode, communication of web respondents with the Blaise database, and security.

1. Introduction

In the fall of 2003 Mathematica Policy Research, Inc. implemented its first true multi-mode Blaise CATI, Web and data entry instrument using one dataset. While Mathematica Policy Research had experience conducting multimode CATI/Web surveys across a combination of software platforms, data were not collected in one dataset in real time. Conducting multimode CATI/Web surveys in different platforms presented challenges such as keeping survey management in sync between the modes, making sure the latest data were in all datasets, maintaining various data-integration scripts, and producing valid reports representing true survey status. The shift to a one-system approach introduced new challenges such as re-thinking security to connect web respondents to the same database used by CATI and data entry. Other issues such as maintaining system performance and Blaise programming techniques are also discussed.

2. Challenges

Implementing the new unified architecture involved reevaluating Web security to meet the new demands of an open link to our internal network and the evaluation and testing of many software platforms. We installed new, more robust hardware to help meet these demands. Additionally, we learned several programming languages and we developed a new philosophy on how the instrument would be constructed and how questions would be displayed for the different modes. Not the least of these challenges involved bringing people together throughout the company to work together on implementing the new architecture.

Our first major step in overcoming this new data collection format was laying out a plan. This meant bringing key people together from each relevant group from the very beginning. This proved to be very helpful as it allowed everybody to become familiar with the concept of our integrated architecture and to speak up if certain known factors could impact other aspects of the project. An excellent example of this was bringing in the network security people at the beginning. As project suggestions were made that could affect security, the network staff was able to address the issue on the spot or suggest better methods of accomplishing the goal. They were able to maintain the right direction security-wise, even as people broke off into subgroups to work on their part of the project

3. Security and Systems Infrastructure

In this section, we briefly discuss the setup of hardware and software infrastructure that was done with the primary goal of ensuring MPR systems security as well as that of respondent data.

3.1 Security

The first major task we had to overcome was how we were going to have one secure dataset accessible from within the company network while allowing Internet respondents access to their data record from this same dataset at the same time. For propriety reasons we cannot go into a lot of details here but will hopefully give you some insight into some of the things we did to overcome these obstacles. The first rule of thumb about security is not to talk about it. All too often people write about their security setup such that users and hackers are given proprietary information or unnecessary clues about the network setup.

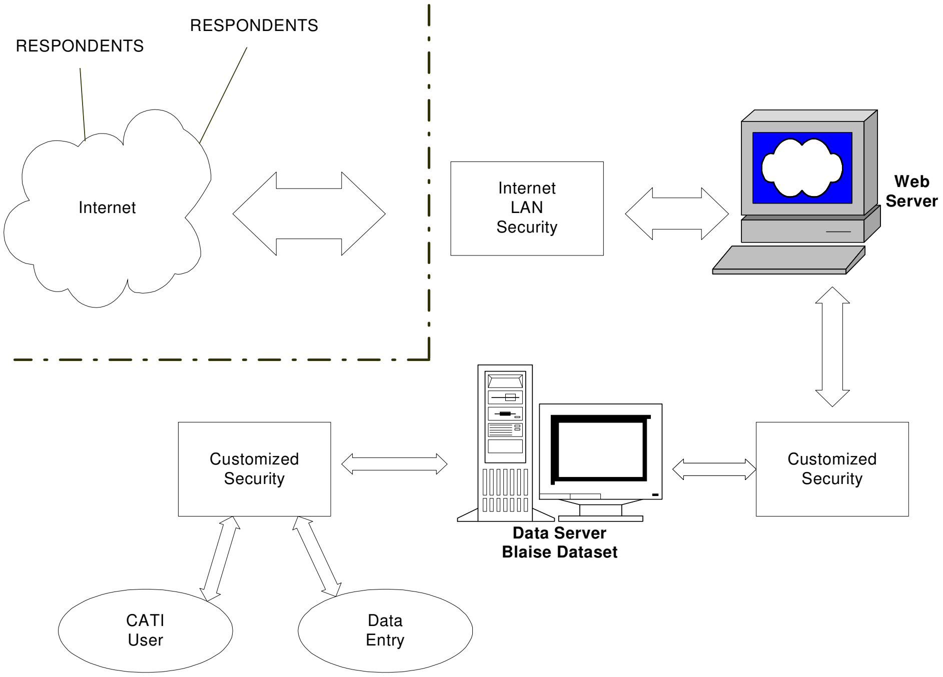
Setting up the new security for the web survey took some thinking and testing, but our past experience with other Web surveys helped us greatly to build a new and more secure system. For instance, we consulted with survey people about characters that should be excluded when generating passwords and how to manage passwords during the web survey. With network people, we coordinated our work for setting up the network configurations, web and data servers and their location, secure port and authentication, and getting the SSL certificate in place. Developers made sure that the data, passwords, and login IDs were kept secure and settings for the web server were correct and locked down. In addition, we decided to delete the first page where the respondents enter their login IDs and passwords before starting the survey so that the respondents or other people would not be able to go back to the first page and reinitiate the survey.

3.2 Hardware

Once the initial security theory was established the next step we took was looking at how the security arrangement would affect performance. This was a major concern considering our security implementation used several pieces of hardware and software to address our security needs. On top of this we wanted a redundant hardware/software solution in case something failed so the survey would still be available. Again we used our past experiences with Web surveys to look at what worked and did not work.

Since the previous Web servers were several years old we upgraded them to the latest technology. The new Web servers had dual processors, dual power supplies and Redundant Arrays of Inexpensive Disks (RAID) hard drive system. Since the price of hardware has decreased greatly the past couple of years we were able to buy some of the latest technology for just a few thousand dollars. The redundancy built into these new servers helped meet part of our goal to have a robust system. However, we knew that the powerful Web servers themselves would not buy us enough speed considering the security we were planning, and so we had to look at other areas. We were able to increase performance by offloading some services onto other servers. We knew this would help but we were concerned about putting more traffic on the network as machines talked with each other. We tackled this problem as a two-step process. The first step was to make sure the communications between the various machines took the shortest route between machines. The second part of this development and probably our biggest speed improvement was installing gigabit connections. We cannot say enough about the gigabit connection and how it increased the speed. The following diagram shows how everything came together.

Figure 1



3.3 Systematic Testing and Deployment Procedures

Once the new hardware, software, and security were in place testing became a very important part of the process. Testing was probably the most critical, ongoing, and multi-phased part of our NSRCG system-management process. It was critical to involve not only technical developers but also subject-area experts from the project in the testing effort. This was true regardless of the degree of specification detail and completeness, as it is the subject-area experts who are in contact with the client and thus who are in the position to best define how the instrument should work and whether any existing written or de facto specifications require modification.

In order to streamline the testing process with minimal production downtime, we had three servers on which we operated: the Alpha, Beta, and production servers. A standard procedure was developed and documented for deploying the updated instrument in each of the three environments, successively. This included not only development of backup and deployment scripts but also of testing scripts for CATI and Web to be used to check and review instrument upgrades simultaneously across modes. Development of basic testing scripts for each mode allowed for efficient and streamlined validation of basic functionality during instrument upgrades.

Once the functionality had been validated and the update approved, procedures were followed for the systematic deployment of any new updated instrument into production. Both initially and throughout, developers experimented with configuration settings on the Alpha server in order to tune performance and modify the interface for the web survey. After developers went through the checklist and tested the web survey on the Alpha server rigorously, the developers then moved the instrument to the mock-production Beta server. On this server, the environment, (e.g., server settings and folder structures) was identical (or nearly so) to that of the production server.

Project personnel could test once the web survey was deployed to the Beta server, and the version that they tested was generally either identical to the one on the production server, or else only one iteration ahead, in preparation for imminent production updates. Our testers performed extensive display and routing testing, as well as the mandatory Section 508 compliance testing. Only after extensive testing was completed, would the developers move the instrument onto the production server. This systematic testing and deployment process minimized redeployment issues, and it prevented critical instrumentation and data problems that could otherwise have occurred. Minimizing downtime is a foremost concern for any survey, but particularly for web surveys that are active 24 hours a day, 7 days a week. Properly implemented and maintained, a systematic testing and deployment process allows for adequate input by the Project testers regarding instrumentation errors or oversight, as well as robust safeguarding with multiple rollback points.

3.4 Browsers

People use many different browsers and different versions of the same browser. There are reports stating that Internet Explorer has about 95% of the browser market but the target group for the NSRCG was recent college graduates. We figured this group of people would be more inclined to use non-IE browsers. Further complicating things was the fact that the sample was international and this meant the possibility of people living in countries where Microsoft products are less widely used. To accommodate these people with the widest varieties of browsers, we tested the instrument using Internet Explorer 5.0 +, Netscape Navigator 4.6 +, Opera, Safari and the newer Netscape and Mozilla/Firefox browser based on Gecko engine. There are hundreds of other browsers in the world but we felt if we address this core group of browsers listed above we would account for 99.99% of the people that have access to the Web.

Part of the HTML syntax for older browsers required some modification but we were able to display the web survey properly for the browsers listed above. To develop a browser-independent web survey was one of our many goals. In order to meet this goal, repetitive and careful rounds of testing across a range of browsers became a standard part of any testing and implementation of new instrument features.

3.5 Software

All three modes of the 2003 NSRCG survey were programmed using Blaise version 4.6 survey software and component pack. For the web survey, we utilized additional software, C2B Internet Service from CentERdata of the University of Tilburg in the Netherlands. C2B Internet Service supports all Blaise features, including don't know and refusal response options, parallel blocks, and Blaise tables. C2B Internet Service is easily configured with both Microsoft IIS and Apache. Using PHP and ASP scripts and C2B commands, the Blaise API was invoked throughout the web survey.

3.6 Languages

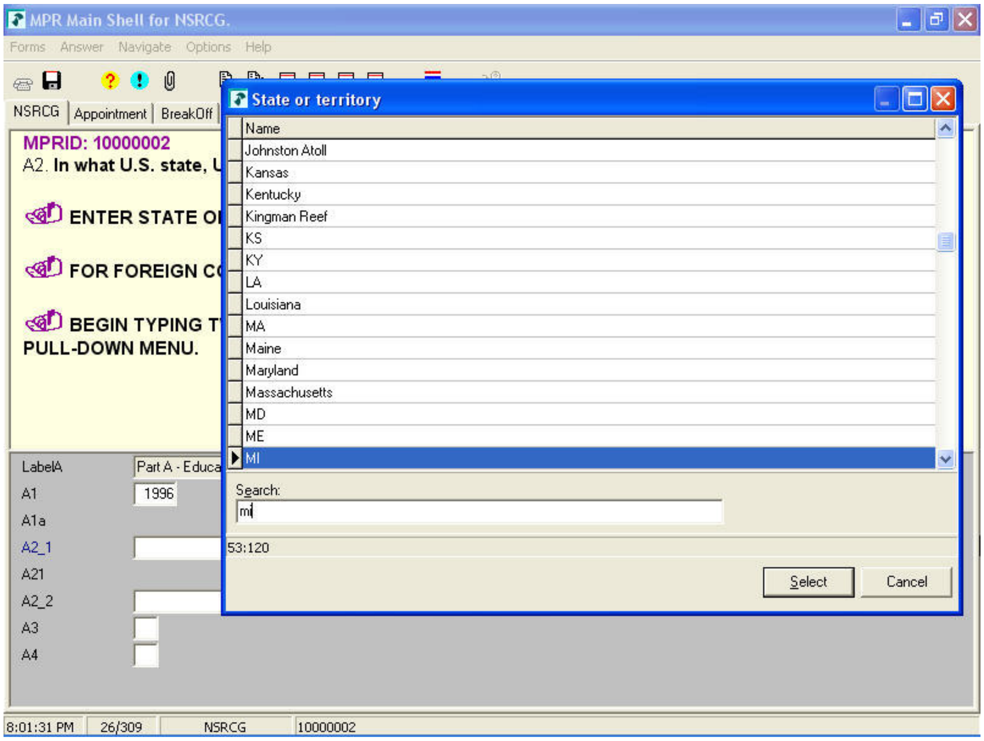
For the 2003 NSRCG web survey, HTML was used in the PHP scripts as well as in many parts of the Blaise source code to display somewhat complicated question layouts properly, including those involving various types of lists and tables. PHP script was mainly used to generate the Blaise questionnaire. JavaScript was also used in the PHP scripts for the client-side scripting. ASP was used for authentication with the LDAP (Lightweight Directory Access Protocol).

4. Programming Issues

There are differences in designing and programming one instrument for multimode CATI, WEB, and data entry, as describe in MPR's companion paper *Instrument Design for a Blaise Multimode Web, CATI, and Paper Survey* by Pierzchala, Wright, Wilson, and Guerino. Here we discuss a few design and programming details that accommodated these differences.

The first-person orientation of the multimode instrument must be kept in mind with regard to questionnaire presentation across modes. For instance, it is very useful for the CATI interviewer and data entry operators to have the capability to use a Blaise lookup table for entering the name of the state (see Figure 2). Interviewers can simply start typing the name of the state and see the lookup table with list of states that starts with the character that the interviewers type. This minimizes typing errors and speeds the interview.

Figure 2



MPRID: 10000002

A2. In what U.S. state, U

ENTER STATE O

FOR FOREIGN C

BEGIN TYPING T PULL-DOWN MENU.

Name
Johnston Atoll
Kansas
Kentucky
Kingman Reef
KS
KY
LA
Louisiana
MA
Maine
Maryland
Massachusetts
MD
ME
MI

Search: mi

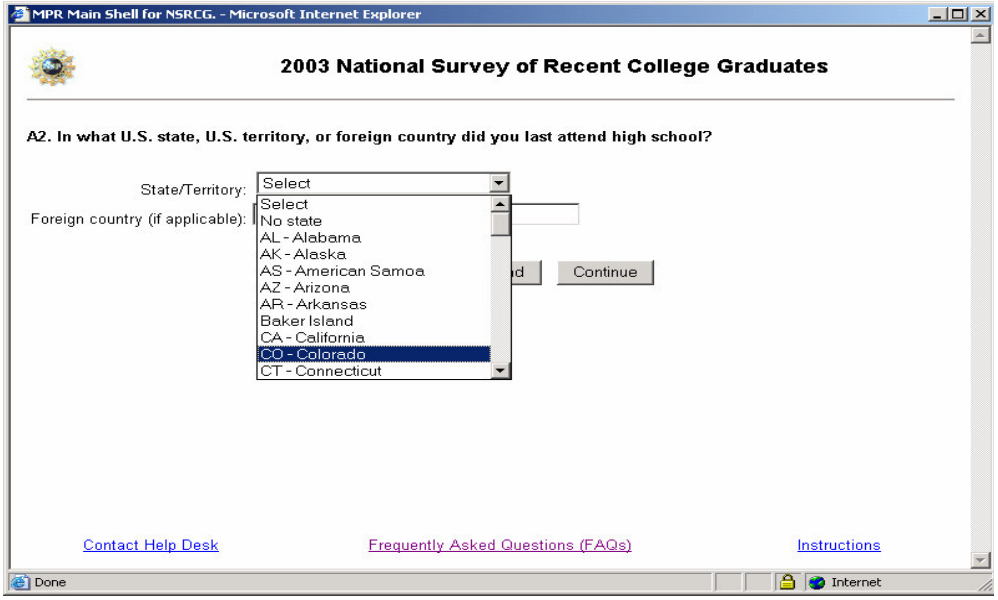
53:120

LabelA	Part A - Educa
A1	1996
A1a	
A2_1	
A21	
A2_2	
A3	
A4	

8:01:31 PM 26/309 NSRCG 10000002

On the other hand, using the drop-down box to display the name of the states to the respondent is more effective in a web survey (see Figure 3).

Figure 3



2003 National Survey of Recent College Graduates

A2. In what U.S. state, U.S. territory, or foreign country did you last attend high school?

State/Territory: Select

Foreign country (if applicable):

Select
No state
AL - Alabama
AK - Alaska
AS - American Samoa
AZ - Arizona
AR - Arkansas
Baker Island
CA - California
CO - Colorado
CT - Connecticut

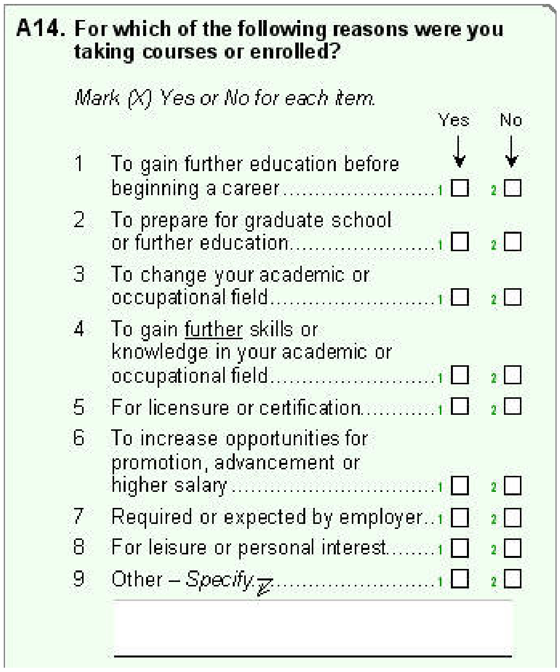
Continue

Contact Help Desk   Frequently Asked Questions (FAQs)   Instructions

The CATI / paper state lookup was accomplished with a standard Blaise external file reference while the same question in the web mode used an enumerated type.

Even in a question that is cast similarly across the three modes, there are also some differences in programming the instrument for CATI and paper entry versus Web. Question A14 was presented as a series of Yes/No sub-items in all three modes. Figure 4 displays the paper version.

Figure 4



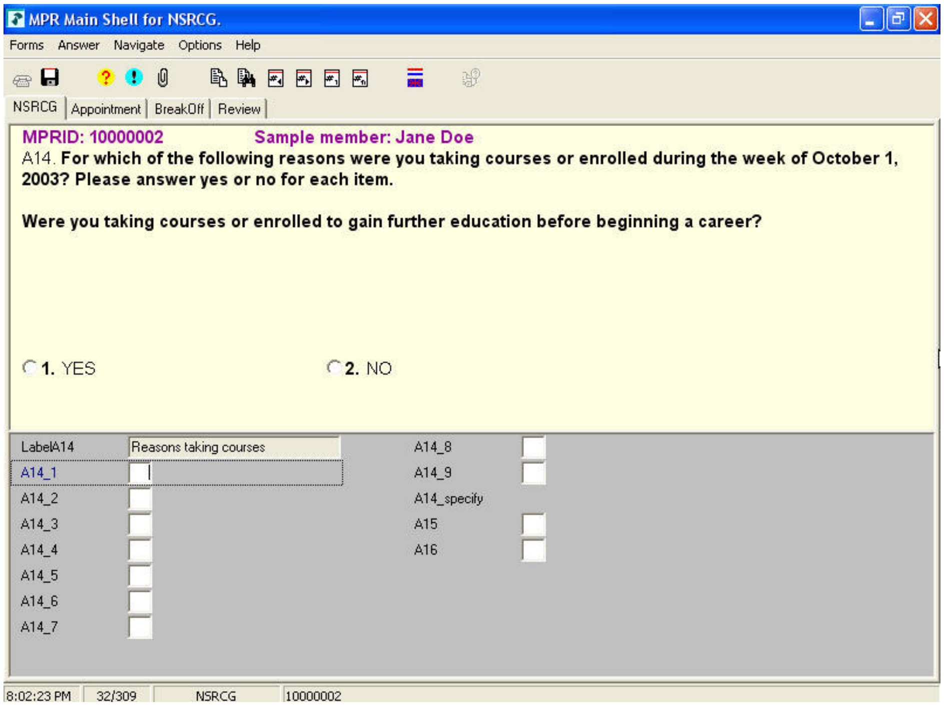
A14. For which of the following reasons were you taking courses or enrolled?

Mark (X) Yes or No for each item.

		Yes	No
1	To gain further education before beginning a career	1 □	2 □
2	To prepare for graduate school or further education	1 □	2 □
3	To change your academic or occupational field	1 □	2 □
4	To gain further skills or knowledge in your academic or occupational field	1 □	2 □
5	For licensure or certification	1 □	2 □
6	To increase opportunities for promotion, advancement or higher salary	1 □	2 □
7	Required or expected by employer	1 □	2 □
8	For leisure or personal interest	1 □	2 □
9	Other – *Specify*	1 □	2 □

In CATI, interviewers read the group of items one at a time to the respondents; the interviewers saw the question texts in succession (Figure 5), but the data entry cells themselves were still displayed on one page of the Blaise Windows interface giving them the context of the complete question.

Figure 5



MPRID: 10000002 Sample member: Jane Doe

A14. **For which of the following reasons were you taking courses or enrolled during the week of October 1, 2003? Please answer yes or no for each item.**

Were you taking courses or enrolled to gain further education before beginning a career?

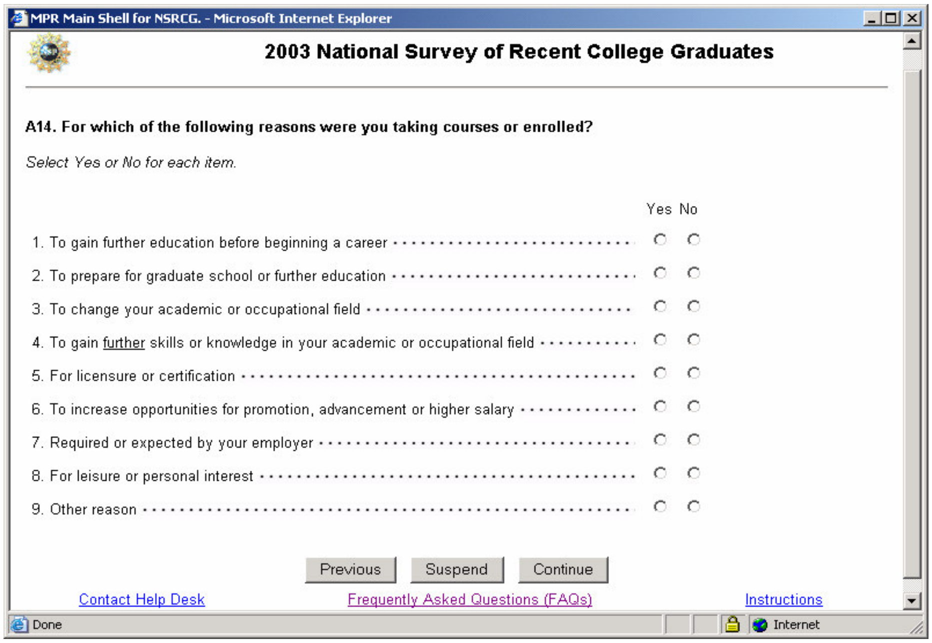
○ **1.** YES ○ **2.** NO

LabelA14	Reasons taking courses	A14_8	□
A14_1	□	A14_9	□
A14_2	□	A14_specify	
A14_3	□	A15	□
A14_4	□	A16	□
A14_5	□		
A14_6	□		
A14_7	□		

8:02:23 PM | 32/309 | NSRCG | 10000002

Web respondents saw the group of questions in one browser page (see Figure 6).

Figure 6



2003 National Survey of Recent College Graduates

A14. For which of the following reasons were you taking courses or enrolled?

Select Yes or No for each item.

	Yes	No
1. To gain further education before beginning a career	○	○
2. To prepare for graduate school or further education	○	○
3. To change your academic or occupational field	○	○
4. To gain further skills or knowledge in your academic or occupational field	○	○
5. For licensure or certification	○	○
6. To increase opportunities for promotion, advancement or higher salary	○	○
7. Required or expected by your employer	○	○
8. For leisure or personal interest	○	○
9. Other reason	○	○

At times, there are noticeable syntax differences in developing the instrument for CATI and Web. For example, to display the first question in the multi-part web question numbered A14 above, the following Blaise question text is used at the block and within-block level:

```
FIELDS {Block-level, used for web browser heading}
  A14 WEB "<b>A14. For which of the following reasons were you taking
           courses or enrolled?</b>
          @/@/@ISelect Yes or No for each item.@I
          C2BTABLE(<table border=0 width = 80%>)
          C2BTD(<td align=center
          valign=top>)C2BNoJava()C2BBlockAsTable()C2BEnumColumns(2)"
      CATI "" : bSectionA14

FIELDS {Item-level within block}
  A14_1 {Education}
    WEB "<table border=0 width=100%><tr valign=center><td>
         <font size=-1><nobr>1.
         To gain further education before beginning a career
         </td><td width=100%
         background=images/dot.jpg></td></tr></table>"

    CATI "@AA14.@A For which of the following reasons were you taking
          courses or enrolled during the week of October 1, 2003?
          Please answer yes or no for each item.
          @/@/Were you taking courses or enrolled to gain further
          education before beginning a career?"
         / "To gain education": tYesNoList
```

We found that, with the current versions of software we were using, the HTML code must be included in the field text for the web survey in order to display the group of items with a nice, legible format. In general, if more than one question needs to be displayed per page in web survey, we needed to embed the HTML code in the field section for formatting purposes. Otherwise, we could simply develop the web instrument just as we would the CATI instrument.

5. Documentation and Testing

As mentioned above, we began with some of the core technical documents required by any project: detailed multimode instrument implementation instruction and testing checklists. A critical part of this core documentation involved further detail on how to set up and configure the C2B Internet Service on our network to successfully host a web survey. After documenting these processes and procedures, we also realized several additional documents were needed and would be vital to the ongoing success of our multimode survey. These critical documents spanned the gamut in their end-user orientation, from testers to developers to call center staff interfacing with respondents. In sum, these documents serve for reference, emergency, and training purposes, as well as baseline documentation for future web and multimode surveys that include a web mode.

Additional necessary documentation included user-oriented instructions for testers on our error tracking application that is accessed through a function key from within the test instrument. Additionally, we found that we as developers had an ongoing need to develop and maintain our technical documentation in greater detail as the survey evolved. Both for general reference and especially as the web commands became more specialized and survey-specific, we found we required additional documentation on how to embed the C2B web interface commands in the Blaise code, as well as detail on and examples of their specific usage. Some extremely useful technical survey management documentation was also added, specifically for the error tracking of reporting items or various general issues reported by either the call center or project management.

Third, and specific to implementation of the web survey mode, we found that fairly explicit documentation was required for the call center Help Desk staff that received trouble calls directly from web survey respondents. This document had to be geared to the respondents' "frequently-asked questions", or, issues of a technical nature that a respondent might encounter, for example, "how do I enable JavaScript?" or "how do I disable the pop-up blocker?" Moreover, the procedures for first-line troubleshooting and implementation of fixes had to be presented in a detailed and stepwise manner for the Help Desk staff. Backup contact information, and the implicit coordination of corresponding after-hours backup technical assistance (covering both instrument environment and network systems support), was also provided.

To properly test the survey across three different modes, one developer and one tester were assigned for each mode of the survey in order to test, fix, and maintain the instrument throughout its current cycle. Each mode of the survey has its unique flow, and thus it was very important that the testers be knowledgeable in such differences and be able to perform the testing in a thorough and efficient manner. Because the source code is shared by all three modes, it is possible to introduce bugs to other modes when the developers fix the source code in one mode. In multimode contexts such as ours, the importance of proper software versioning practices, mode-specific baseline testing scripts, and thorough testing of each new feature cannot be overstated. After making any changes to the code, staff tested the survey in each mode to verify that routing and display are correct for all three modes. In the early part of the survey, testers and developers both spent considerable time taking mock surveys in all three modes (web, CATI, and Data Entry) at the same time to confirm that modes could operate simultaneously. As our procedures and knowledge of the survey matured, it was possible at times to complete testing for an instrument update within an hour or two.

6. Systems Maintenance and Management

The instrument was updated for changes from time to time during live production. When changes were made to the instrument, designated testers conducted thorough testing in all three modes before the instrument was updated in production. While on one hand, updating the instrument for a CATI-only survey can be done before or after the hours of operation at the call center; on the other hand, updating the instrument for a multimode survey with web is not as simple. Respondents can and may be taking the web survey at any time even more so for international respondents. As such, we analyzed the log files from the web server and C2B Internet Service to determine what time was best for interrupting the web survey in order to update the instrument and reboot the servers. We found that web activity was lowest between 2 AM and 7 AM every day; thus we usually updated the instrument early in the morning to avoid interrupting the web respondents. An update that did not require a data definition change could be accomplished in 2 seconds. In this situation, we did not bother to block access to web respondents, we would wait for the Blaise lock file to disappear and would copy 3 files into place virtually instantly. On the other hand, if a data definition change was required, the web access was blocked for a few minutes, with a nice explanatory screen, while the data conversion took place. We also found that activity is usually low on Saturday morning, so we routinely rebooted the servers at that time to clear memory and other resources. Call center staff would log onto a test web case every morning to ensure the web server was running.

We routinely checked that all hardware, including the servers and batteries, were properly maintained and fully functional. As part of this check, we monitored logs for any warnings, as well as the ambient temperature in the server room.

A batch file was run every night to back up important files other than database. The following files are saved in zip files and stored in the designated backup location: PHP scripts, image files used for the web survey, and any log files that are generated by C2B Internet Service and the web server.

7. Other Issues

Call scheduling for CATI was extended to take into account Web logins and completions and cases that were completed on paper and checked in. Cases that were accessed through the web had their record locked in the Blaise database and the Blaise call scheduler automatically excluded them from delivery. If a CATI respondent requested to complete by web, the interviewer would exit the case with the proper notation, and the case would be turned off from the scheduler for a number of days to give the respondent time to finish on the web. Once the interviewer exited the case, it was immediately available for web access to the respondent, including any data already collected. Any complete case, whether completed in CATI, Web, or paper, was denied to the respondent in the web mode. Cases that were started in web, but never completed, were followed up in CATI towards the end of the survey.

Operational, project, and client reports were derived from internal status codes and history files maintained by the instrument. Survey management data dumps were produced every overnight to be available to operational and project staff for investigation of survey management issues. For special investigations, Blaise history and log files, audit trails, C2B log files, and web server logs were used.

Other issues the technical team had to deal with were running an overnight process and linking to *MPR Survey Manager,* an overall survey management system. *MPR Survey Manager* was used for locating, mailings, sending email reminders, checking in paper questionnaires, and other things.

8. Summary

Bringing together three disparate survey modes into one Blaise instrument and database was a challenging technical exercise that was ultimately successful through careful planning, prototyping, and testing. Maintaining open and honest lines of communication between all MPR staff was an essential prerequisite for overcoming a variety of challenges. The fact that security was never compromised, that performance was always maintained in all modes, and that data were collected and processed successfully attests to the effectiveness of the effort.

9. Reference

Pierzchala, Wright, Wilson, and Guerino (2004). *Instrument Design for a Blaise Multimode Web, CATI, and Paper Survey*, Proceedings of the 9th International Blaise Users Conference, Gatineau, Canada: Statistics Canada.