

Index

The Use of Blaise in Organisations

Translation to transformation - lessons learnt and next challenges for ABS Blaise web survey development	3
Experiences in Implementing a Responsive Collection Design (RCD) for Blaise CATI Social Surveys	17
Adapting Blaise and SurveyTrak for the developing world: Case study in Ghana	40
Technical Improvements and future directions for eCollection and multimodal data capture in the Australian Bureau of	49

Case Management

A CAPI Logistic System	62
Lesson learned on preparing and managing multiple modes survey data collection	75
Deploying an Automated Data Read In Process at National Agricultural Statistics Service	87
A Technical Solution for a Mixed-Mode Study	96

Tools

The Michigan Questionnaire Documentation System (MQDS) Version 5 Update.....	106
Research And Development of EHC:The .NET Questionnaire System in EHC.....	114
Using the BCP for Creating SAS Datasets and Codebooks	122
An Improvement of the Application of Blaise CAWI in Verifying Systems of CFPS	131

Miscellaneous

Prototyping Touchscreen Blaise Applications	136
Converting the Labour Force Survey to an electronic questionnaire using Blaise 5.....	149
Blaise 5 Server Configuration for Web Surveys	161

Blaise 5

Tourist Survey from Blaise 4.8 to Blaise 5 in Norway	169
Rendering a Blaise 5 Survey on Different Devices	180
The 2007 Blaise NG White Paper and the Blaise 5 Result	192
Back to the Future: Converting NCVS to Blaise5	201
Developing the UK International Passenger Survey in Blaise 5 on Tablet Computer	226
Studies in Blaise 5	238

Specific survey requirements

Converting the Administration of Health Measures and Bio Specimen Collection from Paper to CAI	242
Programming Blaise for a Multi Questionnaire Study	253
Alternate order of blocks for the CPS-ASEC Instrument	261
Consumer Expenditures Proof of Concept: A Joining of Quarterly and Diary	275

IBUC 2015 Conference Paper

April 2015

From translation to transformation: future challenges and lessons learnt for ABS Blaise online survey program

Author(s): Katrina Martin, Anthony Davies, Christine Harmey, Kathy Buck, Michael Bassett, Julie Larkin.

Presenter(s): Katrina Martin

Organization: Australian Bureau of Statistics (ABS)

1. Abstract

In 2015, the ABS finished a challenging program of migrating data collection activities to online surveys as the primary option for most of its collections. This “Translation” program began rolling out in December 2012 to create new online mode for both household and business surveys, implemented on existing infrastructure. The ABS is now planning for further transformation, with the Acquire capability program due to begin this year. Acquire@ABS will contribute to the ABS transformation goals to modernise and harmonise enterprise wide services, and develop capabilities which will enable the transformation of ABS data collection, acquisition processes and systems. The project aligns with the transformation principles by creating robust, usable and contemporary products and services for providers, external clients and internal clients.

This paper presents work in two parts. Firstly a review of the translation program with successes and lessons learnt as well as recent initiatives beyond migration, including generating Blaise instruments on DDI and utilising the ABS address coder in online forms with an embedded address validation tool. Secondly, looking forward to future online capability, as part of the planned transformation program, Acquire@ABS, and the opportunities and efficiencies sought through improved self response on new digital channels.

2. Background

The ABS is Australia’s national statistical agency. By providing trusted statistics and statistical leadership, the ABS supports public debate and helps Australians to make informed decisions in an increasingly complex world. The ABS provides statistics on a wide range of economic, social, population and environmental matters for government, business and the community. The ABS continues to be regarded as a world leader amongst national statistical agencies. However, the ABS faces a number of challenges, to:

- ensure that high quality official statistics are readily available to governments and the community in time for key decisions to be made about the future
- reduce the cost and time of doing business
- provide open data to users from a range of sources through the provision of modernised statistical products and services.
- reengineer the way we collect data and replace our ageing statistical information management infrastructure.

In response to these challenges, the ABS embarked on a significant journey of change – the ABS 2017 Program, which aims to transform the way we collect, manage and deliver information and statistics. This work started in 2011, and the data collection project is now referred to as the Acquire@ABS project. This project is now one of the 19 identified capability projects under the ABS Critical Statistical Infrastructure Program. The project will develop enterprise capabilities for the data acquisition business model and transformed statistical business lines.

Without this change, our ability to achieve our mission into the future is at risk; we will not be able to continue to fund our existing work program, nor respond effectively to the changing needs of our users. The ABS has applied to government for funding, and in preparation, the ABS has commenced planning for the transformation process.

The ABS began transforming data collection activities to online surveys as the primary mode of collection in 2012; the most significant change to survey collection in the ABS since the introduction of computer assisted interviewing in 2003 and the electronic scanning of paper forms. This was an ambitious and complex change, and many challenges were faced throughout the transition. Initial progress for this translation was presented to IBUC in 2013.

A phased approach was adopted to achieve organisational benefits without additional funding. The first phase “Translation” was established to initiate the ABS 2017 transformation. This stage developed on line survey forms, based directly on the paper forms, delivering essential infrastructure to enable online data collection, and integration with existing workload and collection management systems. This 2 year initial phase, conducted during the 2012-13 and 2013-14 financial years, is now completed, and work is underway to map out requirements for the “Transformation” phase.

Phase 1 outcomes have been met and overall achievements include:

- Cost savings realised for DACB and the ABS with data acquisition and collection
- Delivery of online surveys for a significant amount of business and household collections
- Improved processes applied throughout data acquisition and collection activities for both household and business collections
- Providing an electronic interaction mode for data providers
- Meeting whole of government requirement to reduce provider burden and increase electronic interaction with the public
- Delivery of a digital Census test
- Facilitating and adopting a collaborative approach to change management throughout the ABS.

3. Current Blaise web program

In 2012, Blaise IS was assessed as a suitable platform for current and future survey collection requirements, offering a ready-made solution that would integrate well with existing survey collections. This solution needed to be available for deployment for the MPS from the December 2012 cycle. The ABS has since implemented online forms for the majority of business collections and the Monthly Population Survey (MPS), completing the first phase of the ABS data collection transformation, with:

- upwards of 90% being collected via web for Main Economic Indicator (MEI) quarterly business surveys
- self-reporting take-up for on average 22% of MPS sample.

Business surveys enjoy a high take up rate as these collections were always self enumerated, and the providers were demanding online capture capability. Take up rates for the MPS online form is much lower. Household surveys traditionally involve interviewer modes of data collection including obtaining data directly by way of telephone interview (TI) or face to face (F2F) interviews.

3.1. Business online form progress

Business respondents initial take up for online surveys were extremely good, and the rates are still, in general, increasing over time. This is primarily due to meeting provider expectations to have a digital channel. The transition has been very successful for most surveys, especially for the straightforward self enumerated paper form. Collections of these types were a 1-1 map into a Blaise online form. When surveys had a more complicated path the translation into an online instrument and acceptance by providers was less successful. For example, the Blaise Computer Assisted Telephone Interviewing (CATI) instrument is deemed more appropriate for the Job Vacancies Survey for internal collection, than a respondent logging on to an online form. Providers associated with Agricultural surveys (traditionally long with complex response categories) have been found to be less likely to convert to online form collection, although take up rates for Agricultural surveys has improved from 2013 to 2014.

3.1.1 Operational benefits

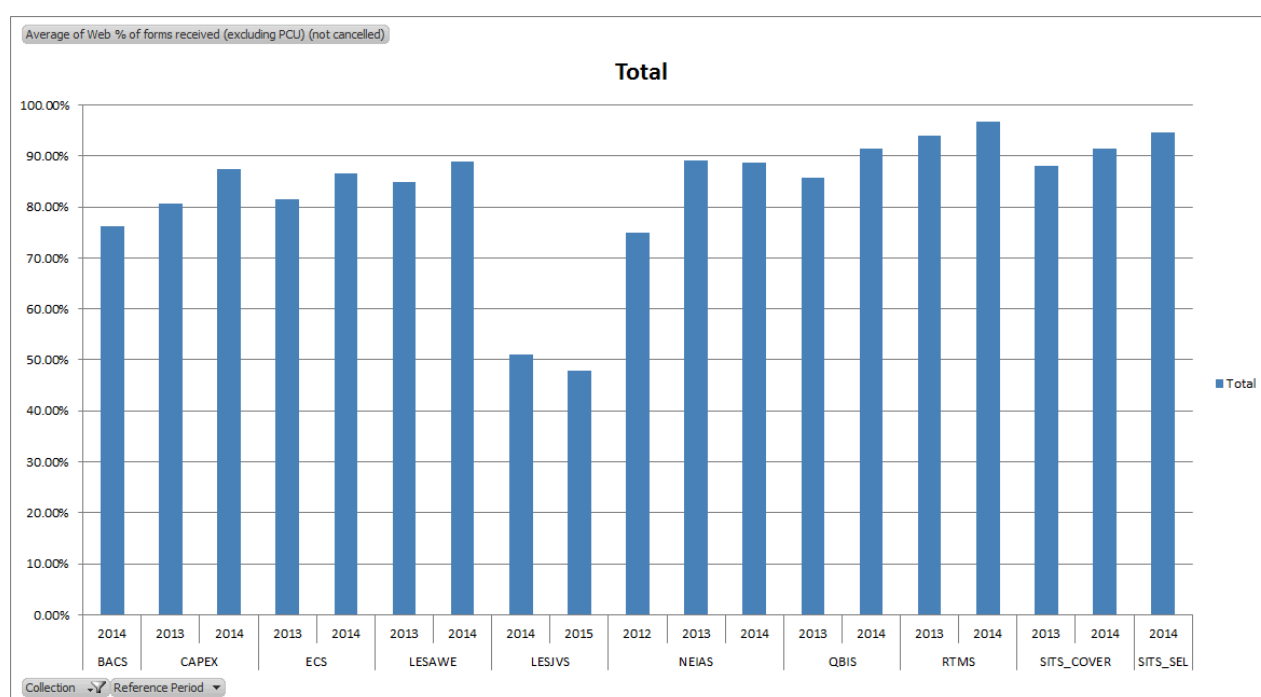
The operational benefits from transitioning the majority of business forms to online surveys are significant. The reduced number of paper forms despatched, with corresponding savings from less paper form data capture and processing, is predicted to save the ABS approximately 2.4 million per year.

However, additional operational costs have been incurred, with significant support required in relation to password resets (including forgotten passwords, logon difficulties, discarded log-in information). Work is currently underway to changing means of authentication to promote self managed passwords (more in section 5.3 below).

3.1.2 Business take up rates

The following table shows online form take up rates for a range of business collections from December 2012 through to March 2015.

ABS quarterly Business Surveys - Average percent by year

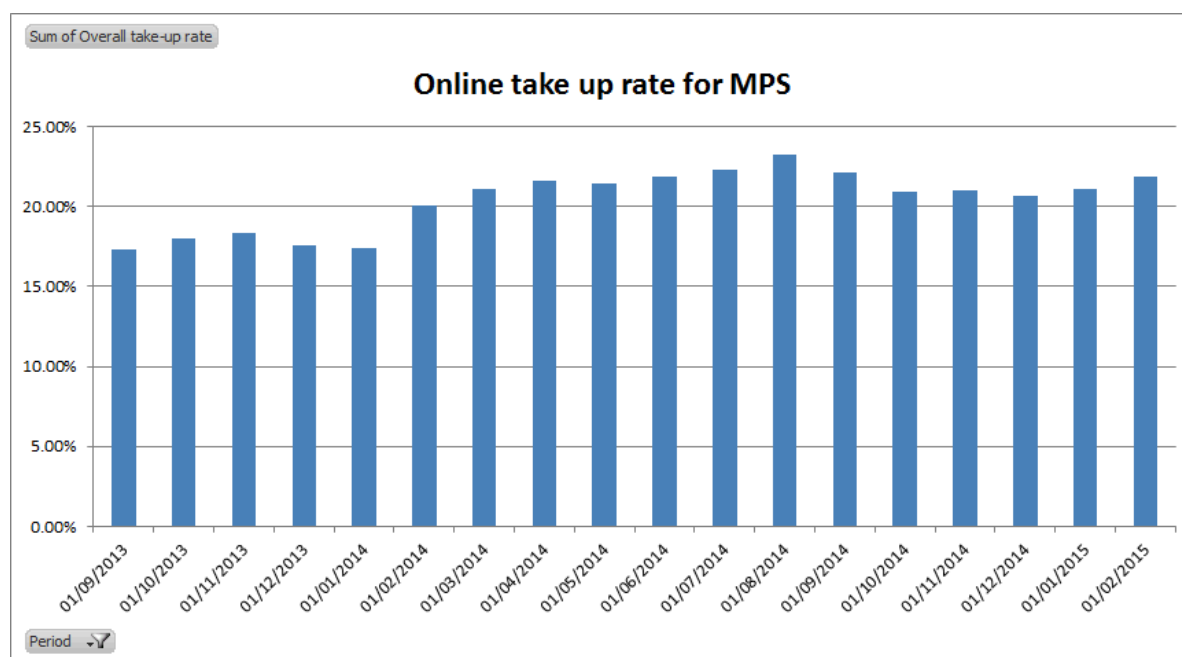


3.2. Household online form progress

The Blaise household contact detail form was introduced in 2012 to reduce interviewer travel costs when attempting to make contact with a dwelling. This solution was implemented using a simple Blaise online form, to capture the preferred contact times and numbers for interviewers to call. This had moderate success due to minimal take up, but the operational gains from when it was used were significant.

The MPS moved to an online form in late 2012, with the initial web option only available to subsample of total MPS population before gradually increasing availability in mid 2013 to full sample. Take up rates have increased, hovering at about 21- 22%. The drop off rate for online form reporting has influenced these take up rates. It should be noted that an online form reporting option is re-offered in the fourth month of participation but was not promoted in earlier periods.

MPS online survey take up rates – December 2012 to February 2015



3.3. Online Census Form progress

Due to scalability requirements, the online census form is not using the current ABS Blaise environment. The ABS is reusing the eCollection solution from 2006 and 2011, as developed by IBM. The recent Census Major Test conducted in August 2014, had 57% of responding dwellings used the online option and the remainder used a paper form, a promising improvement on the 33% of Australian households who chose to complete the online survey during the 2011 Census of Population and Housing. This increase was driven by a combination of factors, including increasing use of the internet by Australians to interact with Government between 2011 and 2014. The online option was promoted as the preferred option in the approach letter sent to most households. Paper forms were primarily only sent out by request with a small proportion receiving them up front. This growth in take-up is encouraging and bodes well for the planned 65% online take up for 2016.

The online form developed for the 2014 Census Major Test were designed and built to be responsive. This is the first time the ABS has included responsive design for eCollection. Responsive design enables the form layout to adapt based on the screen size of the respondents device. Adjusting the layout improves the usability of the form and therefore the respondent's experience. As part of

meeting provider expectations, the ABS is including responsive design as key requirement for future household and business online surveys.

3.4. Usability and design

The current platform has been designed for desktop computer users, primarily for Internet Explorer browsers and Windows 7, which accounts for nearly 50% of respondents. The online forms are tested across a range of browsers to make sure behaviour is as expected, as variations can occur, mainly around screen layout of response boxes. The results for household and business collections are presented separately due to the differences in device usage. Respondents do use mobile devices (Android and iOS), however at present, ABS instruments are not optimised for said devices, which has led to some respondent dissatisfaction. The low numbers in the tables below reflect the difficulty of completing a form through handheld devices, not preferred respondent options. Improvements for responsive design are limited in our current solution and will be included with future Blaise developments or alternative platforms.

Business online survey responses by platform and operating system

Browser	Operating System							
	Android	iPad	iPhone	Mac	Windows	X Window	Other	Grand Total
Chrome	0.11%	0.04%	0.00%	0.68%	17.27%	0.04%	0.00%	18.15%
Firefox	0.01%	0.00%	0.00%	0.66%	8.87%	0.03%	0.00%	9.58%
Internet Explorer	0.00%	0.00%	0.00%	0.00%	66.43%	0.00%	0.01%	66.44%
Opera	0.00%	0.00%	0.00%	0.00%	0.02%	0.00%	0.00%	0.02%
Safari	0.17%	1.63%	0.33%	3.23%	0.43%	0.00%	0.00%	5.79%
Other	0.00%	0.00%	0.00%	0.00%	0.02%	0.00%	0.00%	0.03%
Grand Total	0.29%	1.67%	0.33%	4.58%	93.05%	0.07%	0.01%	100.00%

Household online survey responses by platform and operating system

Browser	Operating System							
	Android	iPad	iPhone	Mac	Windows	X Window	Other	Grand Total
Chrome	1.02%	0.22%	0.04%	2.21%	22.79%	0.04%	0.00%	26.30%
Firefox	0.14%	0.00%	0.00%	1.60%	12.57%	0.30%	0.00%	14.61%
Internet Explorer	0.00%	0.00%	0.00%	0.00%	37.38%	0.00%	0.07%	37.45%
Opera	0.00%	0.00%	0.00%	0.00%	0.05%	0.00%	0.00%	0.05%
Safari	1.00%	9.59%	2.22%	8.55%	0.09%	0.00%	0.00%	21.46%
Other	0.00%	0.00%	0.00%	0.00%	0.11%	0.02%	0.00%	0.13%
Grand Total	2.17%	9.81%	2.26%	12.35%	72.98%	0.36%	0.07%	100.00%

3.5. Online form accessibility

The ABS has recently undertaken a review on current Blaise online forms and the WCAG (Web Content Accessibility Guidelines) standards. This included how suited the forms are for those respondents with vision impairment, or who rely upon screen readers like JAWS or NVDA. While not fully compliant (at present due to limited resources), most issues identified within the current form design should be straightforward to resolve. For example, simple aspects like using alt attributes on images, or a consistency with marking up headings, and others around how Blaise has rendered in to HTML (eg. marking up table cells). The ABS is currently working on remedying these aspects but major changes will not be made until further transformation projects are undertaken.

4. Reviews to support Blaise online form implementation

The ABS largely self-funded the initial web migration program, with internal governance to oversee the program across the many stakeholder groups. Several key reviews were held while the translation projects were underway. This paper reports on key outcomes from the sprint workshops to improve efficiencies in operations, and the review of Blaise development process.

4.1. Sprint workshop outcomes

Workshops were held in business and household collection area stakeholders to develop strategies that were jointly agreed to be of strategic importance, outlining challenges and highlighting the need and the importance of working in partnership with stakeholders to achieve the best outcomes. In particular: the budget context of increased efficiency dividends and the need for productivity initiatives; the need to produce quality data and 'fit for purpose' statistics for our users; the ABS' outline of stronger accountability for financial management; and the ABS2017 goal of reducing cost and time of doing our business.

4.1.1 Reduced data collection costs

Significant savings have been realised through reduced printing and data capture costs with the introduction of online forms. While online forms are now the primary mode for most business surveys, follow up of these surveys rely on paper letters and telephone - both of which are expensive follow-up modes. An Email Reminders functionality was identified as a priority initiative for business surveys to deliver additional benefits. A new project was established to implement Electronic Reminder Application (ERA), and meet business survey respondent needs to interact with Government through electronic channels.

This is a larger scale development project than households, due to the size of the ongoing business program. The ERA work has established an architectural design, new business processes, and communication strategy for optimising the outcomes from the digital channel. Pilot tests across the business respondents have been positive, with a general result that those selected for email reminders had a much higher rate of completing survey, and those who provide emails in online form contact details are more likely to complete the survey in the next collection cycle. The functionality is due to be released into production by end June 2015.

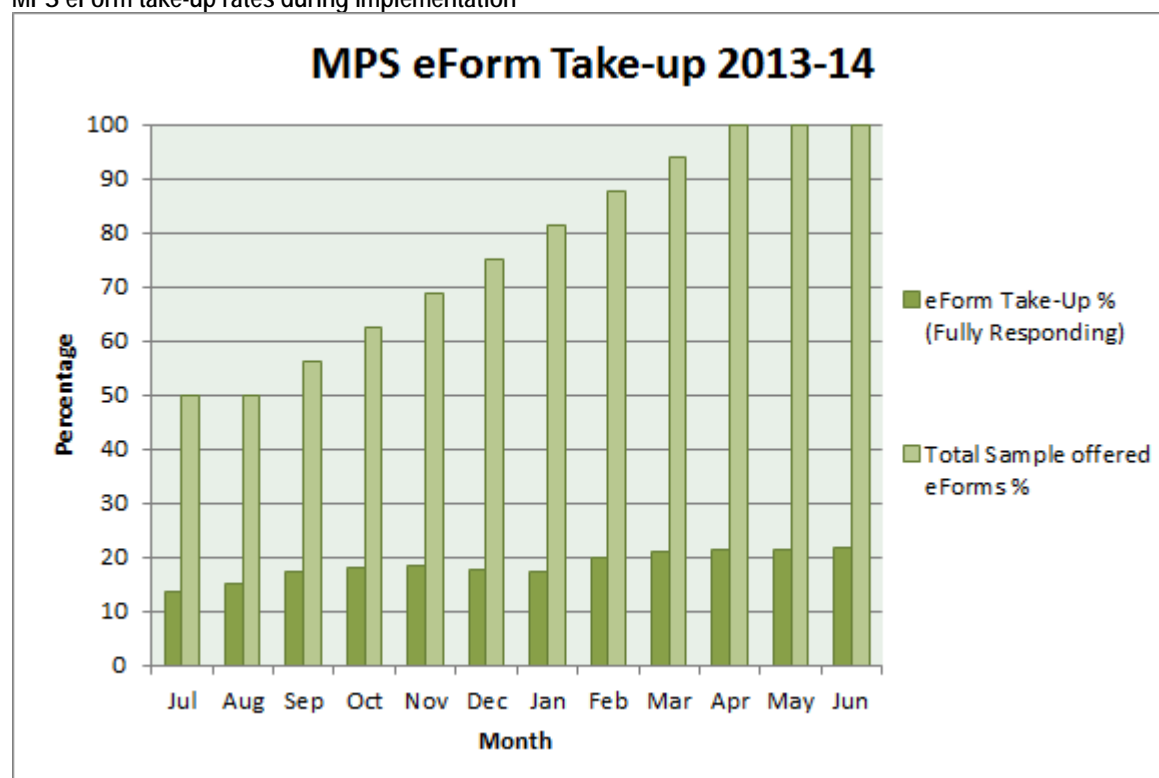
4.1.2 Improved online form take-up

Clearer and more focused approach and reminder letters have been created for business and household surveys. The new letters are designed to improve the take-up of online form and the call to action to providers. A3 fold-out colour brochure, referred to as the Guide, created for household pre-approach information, including MPS. These initiatives saw an improved take up of 3-5% to the online form.

Strategy changes were endorsed by senior Executive to strengthen the compulsion statement in correspondence, and for letters to be sent with the Australian Statistician's signature block. The new compulsion statements were tested in the Health dress rehearsal in November 2013 and went live in February 2014 for MPS enumeration, with another increase of 2% take up. Business surveys have also implemented this change from the December 2013 quarter. The compulsion statements clearly state that respondents are obliged to provide data to the ABS.

In addition, all MPS online enumeration was shifted into week 1, to improve the follow-up process, providing particular benefit to the enumeration of supplementary surveys. This was also implemented in February 2014, and adds to the take up rates displayed in the table below.

MPS eForm take-up rates during implementation



4.2. Blaise development review

The ABS held a review in early 2014, after the majority of online forms were implemented, to assess the impact from the new work on the statistical and development business lines. The project identified the changes introduced to the data collect processes, and the coordination necessary to allocate resources effectively and deliver the entire survey program efficiently.

4.2.1 Blaise review outcomes

The aims of the Blaise review were to:

1. improve the Blaise instrument development process by addressing practices leading to non-sustainable resource stretch, timeliness and quality issues
2. identify, prioritise and address issues leading to immediate improvements and relieving pressures experienced during development of collection instruments
3. integrate changes to the Blaise instrument development process and timing to accommodate multi-modal instrument design
4. standardise and streamline processes across business and household surveys to achieve efficiencies, improve quality and provide consistency for providers
5. provide best practice guidelines and standards in a consolidated process.

A period of consolidation for the last 6 months of the project gave opportunity to bed down the online form surveys. This opportunity was also used to review processes involved in survey instrument specification, development and testing with a view to identify and address pain points in the Blaise instrument development processes; implementing improvements to relieve identified pressures; standardising and streamlining processes across business and household surveys and accommodating multi modal instrument design.

4.2.2 Lessons learnt

Transforming data collection to online forms, while delivering on collection operations, was a demanding undertaking that placed pressure on many areas of the ABS. Statistical areas were contributing to the development of the new online form stream while maintaining ongoing operations. Some strategic risks were realised during the project and required remediation, particularly from system instability and poor performance arising from legacy system integration. The additional complexity from the new channel addition has increased operations and administrative load, with significant impact on available resources.

The review was undertaken in close collaboration with household and business survey areas, the Blaise development team, and other key stakeholders, and identified the following key themes:

- A provider focus is required when developing transformed solutions
- The rapidly changing digital world imposes challenges (devices, platforms, online environment)
- Integration of new functionality (online data collection mode) with current legacy systems is complex and difficult.
- Don't underestimate complexity of delivering change across the organisation. Change management practices are as essential as sound project management to deliver quality outcomes for large projects.
- Support across ABS from senior Executives was important. Many challenges were faced in transition.
- Benefits realisation plan to be developed at project outset and allow for impact of transition (including how benefits will be measured).
- Need to resource implementation, including for SMAs to adopt and implement change. Process improvements are essential to support system changes to achieve effective outcomes.
- Future change solutions to meet enterprise capability requirements.
- Development work to be done in separate stream to business lines, good engagement essential for good outcomes.

Changes to survey program management from the Blaise review are underway, with benefits realised almost immediately through improved communication and collaboration between stakeholders. Experience gained through the initial migration process has led to improved timetabling and resource estimation for upcoming developments. Standardisation of the code base is being undertaken by the Blaise team. Overall, ABS online survey deployment capability has improved, demonstrated by better quality instruments and lower overall costs (i.e. development, testing and corrections).

The next phase of Acquire will move from operational delivery of working web form capability to providing the enterprise capabilities for data acquisition to support reengineered statistical programs and data acquisition business model. That data collection is going first reflects the priority to replace the aging infrastructure currently supporting the Blaise web platform. The online survey infrastructure in place now was retrofitted to the existing paper/interviewer based systems and the operational areas have worn the costs of this, mostly with additional human intervention to manually support the modified processes.

5. Additional enhancements to online form environment

The ABS introduced additional improvements in the translation program that were beyond the scope of the original aim for replicating the paper questionnaire in an online format. These projects were introduced as capabilities to secure the infrastructure environment (as needed to manage risk from authentication errors) or externally funded to support specific survey goals, such as the metadata instrument generation for Agricultural surveys.

5.1. Generating a Blaise instrument with DDI

During 2013 the ABS developed a prototype metadata driven Blaise online form. This involved developing a Metadata Authoring Tool instrument design interface to capture the questionnaire specifications in DDI. The survey instrument specification would then be automatically transformed into Blaise code, forming the foundation of an online form.

The prototype was designed to use strategic ABS technologies, platforms and tools, including the Statistical Workflow Management (SWM) and the Metadata Registry and Repository (MRR), as well standards such as GSIM, GSBPM, and DDI to support automation of statistical business processes for online form production.

The project was successful in enabling automatic generation of Blaise online form code from a DDI instrument specification was achieved. Approximately 80% of Blaise online form code was automatically generated and used as the foundation for the 2013-14 Land Management Practices Survey (LAMPS) resulting in a reduction in development time from 10 days down to 2 days. Furthermore, the project also laid the foundation for developing the ABS DDI Instrument Profile based on the DDI sample that was produced for the LaMPS instrument. Finally, the project demonstrated the use of the Metadata Authoring Tool to define instrument specification in DDI in an efficient manner to enable potential streamlining and automation of online form development in the future.

5.2. Address validation tool

The address validation tool prototype was developed for the 2014 Agricultural collections, to improve address quality and geocoding outcomes. This was the first instance of embedding functionality within the Blaise form to target the respondent checking entered information, with intention of reducing downstream coding efforts.

The tool uses the ABS address validation to confirm the address and then captures the geocode information for the validated address from the ABS address coder. The level of geocode information written to Blaise can be specified to meet business needs. The first instance operated on a text basis, asking respondents to confirm an entered address against a known list of addresses presented to the user from an internal address register. The prototype investigations included map functionality, however the additional web calls were determined to be too heavy. Future development will consider how to reduce this to support user interaction from a graphical map interface, but may not progress until further transformation programs are initiated.

5.3. New authentication solution

The rapid transition to online enumeration exposed an increased risk of data disclosure, when providers called the contact centre for help to access their online form. The instability of the first authentication solution and the potential for human error in verifying the correct business did result in accidental disclosure of active online forms. The ABS has improved the verification process and is now implementing a new authentication solution, for use by end June 2015.

The new authentication solution, referred to in-house as XIAM (external identity access management) replaces the first solution, where the authentication process was embedded within the Blaise environment. The XIAM solution utilises off-the-shelf software for password management, to have respondents self-register and establish credentials for ongoing management of their access codes. This is an essential component, to give users the tools to self-manage passwords, and reduce the number of incoming calls to the contact centre requesting a password reset.

The XIAM solution will be the new corporate capability for registration. The environment has benefits for supporting online collection, especially for managing upcoming selections for respondents as sets of obligations, i.e.: only one access point required for all 8 months of the MPS. Additionally, the new environment allows for use of multiple server parks, giving more flexibility for introducing updated Blaise versions, and supporting different online collection platforms. The solution should be readily extendible to meet Whole-of-Government alignment with the 'one government' access portal, such as 'MyGov' and 'Auskey'.

Considerable care was taken in the design of the overall solution, as the user navigates through three distinct systems. The initial log in screens, registration pages and survey obligation pages were designed to have a consistent look and feel, and make it easy for respondents to use. The increased registration time is being assessed to minimise impact on household respondents particularly. Initial testing is positive, at time of writing the ABS were preparing for a large scale pilot test (1000+ sample) across household respondents. The key measure of success for the June go-live date will be reduced number of calls through to the call centre to reset passwords.

5.4. Additional research projects

The ABS is progressing additional research projects to improve provider experience and encourage self-reporting. These enhancements focus on increasing the household online survey responses for the MPS, where small conversion rates give large benefits through reduced field costs.

5.4.1 Online conversion

MPS survey is conducted monthly over a 8 month period. Currently, interviewers attempt to convert respondents into online reporting in the first and fourth month of data collection. Proposed procedural changes are underway to attempt conversion in consecutive months from CATI/CAPI to CAWI in the first to fifth months and offering the online survey every month via letter and email to all respondents that have completed an online form. These procedures aim to shift providers to use online forms and further increase the take up rates.

5.4.2 Making online entry visible

The providers are sent a letter with a shortened URL to go directly to the log in screen. However it was noted in testing that providers may search on the website or through browser search engines for the log in screen. To accommodate for this, the ABS website now displays a 'complete your survey' button which enables respondents' direct access to log into their survey obligation.

5.4.3 Form redesign

The Blaise household contact detail form was introduced in 2012 to reduce interviewer travel costs when attempting to make contact with a dwelling. This form will be redesigned to increase a positive experience for the respondent. Respondents can mistake this form as the actual survey form and be disgruntled when they realise they have to complete another form. The Blaise household contact form will maintain its autonomy by presenting the online form as one page without 'next/back' buttons. The redesign aims to present it as a page for information which is separate to the survey form.

6. The journey to transformation

The future transformation of online survey forms is part of the Acquire@ABS project, the first of nineteen Enterprise Wide Capability Projects to be delivered under the ABS transformation program CSIP (Critical Statistical Infrastructure Program). CSIP seeks to ensure a sustainable future for the ABS. The program will implement large scale innovation by standardising and industrialising ABS business capabilities and driving these automatically through the use of end-to-end metadata. This will require the re-engineering of a large number (if not all) statistical business services. For this to be achieved, a number of CSIP projects will be initiated where requirements will be gathered, services built, bought or wrapped (whereby existing services will be incorporated into new environment) and business activities integrated with the new foundational infrastructure.

6.1. Building capability

Acquire@ABS is an enabling project that will transform our data acquisition capabilities through harmonised enterprise wide services which will enable the transformation of ABS data collection, acquisition processes and systems. The project aligns with the ABS 2017 principles by creating robust, usable and contemporary products and services for providers, external clients and internal clients.

6.1.1 Aims for Acquire project

The Acquire@ABS project aims to integrated enterprise wide solutions to transform the way we collect, manage and deliver information, and is tightly aligned with three ABS Strategic Directions:

- Transformation of the management of statistical information to improve usability, value and timeliness
- Meet provider expectations through greater coordination of information collected for statistical purposes and through investment in e-collection and other alternative data sources
- Ensuring long term sustainability by developing scenarios that maximise the effective and efficient use of our resources.

6.1.2 Enterprise solutions

The ABS aims to move from line of business infrastructure to enterprise infrastructure, which is able to meet the highest priority needs of most areas. Both the Acquire@ABS and Census programs will be responsible for enterprise infrastructure development for the collection of data over the next few years, with the aim that for 2021 the collect environment will have capability to support both the complexity of the ongoing survey program and the magnitude of the population census.

As discussed, to date the ABS has focused on implementing online survey capability for the Monthly Population Survey, most business surveys and preparation of an online solution for the Census in August 2016. Due to issues of scale, the Census solution is not being considered on Blaise for the 2016 cycle. However many components in development for the 2016 Census infrastructure can support ongoing online collection.

The enterprise solutions will provide the infrastructure to support the new transformed business processes, and use enterprise architecture through the following principles:

- Enterprise infrastructure such as the MRR should underpin new developments, and all systems at a minimum need to be metadata driven.
- A common look and feel should also be maintained where possible and appropriate.
- Where practical solutions will be developed following Service Oriented Architecture/ Modular design (plug and play) principles to maximise reuse and reduce ongoing support.
- Leverage for future corporate solutions. What is underway from 2016 Census solutions. Online Census tool externally built and hosted application not built for future survey program

re-use. Investigating expanding application solution now to extend re-usability. Note 2011 ACES solution had similar basis and not supported for future viable solution.

6.1.3 Future business needs

Online instrument capability is a core part of all future collection strategies. Data collection will be primarily self-reporting, through digital channels. The future pathway for reengineered population statistics is for a new household survey, the Australian Population Survey (APS), to integrate previously disparate survey data collection into a single, integrated, continuous modular vehicle. This will enable increased efficiency, responsive agility, statistical utility and data coherence.

Business data transformation has a similar emphasis on coherence, to align collected data with administrative data and minimise reporting burden. Central to this is an expanded and outwardly facing Business Register, and unlocking the potential of administrative and transactional data. Whole of government business engagement initiatives will be supported, including use of AusKey for survey authentication, ability to access ABS surveys through “business.gov.au” and prefilling forms with Standard Business Reporting data already collected by other agencies.

Blaise remains in consideration as a part of this strategy however where it fits is dependent of how Blaise 5 will work with the desired outcomes: a flexible and adaptable environment enabling a more responsive statistical system.

6.2. Plans for future data collection

Acquire@ABS has evolved over time to meet the challenges presented to deliver a web collection capability and infrastructure that is convenient to providers and respondents. Through Acquire@ABS, data collection will transform to enable business areas to modernise processes, with systems which are easy to use and reliable for internal and external stakeholders. Acquire@ABS will create a contemporary environment for collection and communication with our providers, clients and workforce.

This transformation will support the collection or acquisition of data via a range of capabilities and activities. This includes statistical data requests which enable data providers to be notified of their need to provide data to the ABS. This capability includes the construction of data collection instruments to support different modes of collection (e.g. web queries, file transfer, self administered/other interview type, etc.), and notification using different channels (e.g. electronically, phone, physical mail). Key capabilities included in the project are:

- Instrument creation through metadata driven questionnaire design tool
- Multi-modal data capture and processes
- Clerical data capture
- Follow up of information from data providers and data collectors

This project enables the retrieval of statistical data from survey, census, or administrative data sources via a range of data channels including interviewing, web capture, scanning, file transfer and stream transfer. The ABS is aiming to have a build once use many approach, where instruments are specified once and instances for modes generated via metadata driven process to support dynamic collection activities.

The ABS is currently investigating approaches to market to create this capability, as previous approaches have indicated large gaps in the currently available off the shelf products. This approach may consider multiple platforms for online capability, to support simple and complex questionnaire requirements. The ABS is engaging with other NSOs to consider how international collaboration may provide benefits with a market solution. In addition, the ABS is considering how to best leverage current capabilities to meet future needs, as discussed in section 6.1.2 above.

6.3. Provider experience

The ABS collects data under the Census and Statistics Act, which obliges respondents to provide data as requested and protects the confidentiality of the entity providing that information. This assists the ABS to maintain a high level of compliance, achieved through extensive face to face and intensive follow up (IFU) telephone procedures. Engaged providers give data more willingly and need less ABS follow up requests. Future collection operations seek to benefit from higher self-response.

In simple terms the ABS of the future aims for engagement that supports self-reporting needs, ie: simple, fast and secure, such that our providers will say:

- “I trust the ABS to look after the information I give them, and understand why they need it.”
- “It is easy to report my data to the ABS when I am asked.”
- “The ABS values my time and respects my needs. I feel my efforts are appreciated.”

While the ABS has made strong progress to date in building on-line capability, the quick wins for improving response are nearly exhausted. There are good opportunities for managing provider reporting. However, aging infrastructure will not support increasing on-line reporting and severely limits future engagement options. The ABS 2017 transformation program aims to:

- implement provider-centric systems, to support tailored interactions and improve the provider experience
- build capability for responsive design, starting with collecting information on provider behaviours and preferences
- consider marketing and commercial approaches, such as CRM systems (Customer Relationship Management platforms), to track, record, store, and then data mine the information in a way that increases customer relations
- use provider intelligence for identifying non-compliance clusters
- build potential for future profiling of big data, individualising collection approaches and offering omnibus surveys for households and businesses.

6.4. International collaboration

The ABS is actively working with other National Statistical Organisations, to identify collaboration opportunities to support transformed collection infrastructure. This engagement includes working with Statistics Netherlands to progress priorities for Blaise 5 to support the digital by design, multi-modal collection approach outlined above. Considerations include how to assist the development program through shared resources, testing environments, performance and load capacities.

7. Summary

The ABS has completed the translation work through provision of online surveys for the majority of business collections, the Monthly Population Survey and provision of an online form for the Census Major test in August 2014. The ABS introduced several review points to improve the operational management of the new processes and increase uptake of online forms by examining and improving pre-approach and follow up strategies for households and businesses.

The ABS is now preparing for transformation, while waiting for a funding decision from Government. The transformation will revolutionise how the ABS work, standardising and industrialising business capabilities and driving these automatically through the use of end-to-end metadata. For data collection, this will build on the gains already made through the translation project. The new environment will be provider-focused, to make it easy for self-reporting, with online collection the primary mode. This capability needed is to support the new digital channel, with metadata driven questionnaire tools, online smart form functionality to reduce downstream editing, responsive design to support mobile devices and systems that are tuneable for the complexity and scale necessary to support the diverse collection program of a National Statistical Agency.

8. References

ABS Annual Report, 2013-14 (Cat no. 1001.0)

[1001.0 - Australian Bureau of Statistics -- Annual Report, 2013-14](#)

ABS Corporate Plan , Jul 2012 (Cat no. 1005.0)

[1005.0 - ABS Corporate Plan, Jul 2012](#)

ABS Forward Work Program 2012-13 to 2015-16 (Cat no. 1006.0)

[1006.0 - Forward Work Program, 2013-14 to 2017-18](#)

IBUC 2015 technical paper: Technical Improvements and future directions for eCollection and multimodal data capture in the ABS

MSIS 2014 paper: Statistical metadata driven eForms [Link to paper](#)

IBUC 2013 paper: Challenges of migrating ABS business and household surveys to Blaise web on a large scale and in a short timeframe [Link to paper](#)

Experiences in Implementing a Responsive Collection Design (RCD) for Blaise CATI Social Surveys

Éric Joyal, François Laflamme, Statistics Canada

Over the past few years, paradata research has focused on gaining a better understanding of data collection processes, leading to the identification of strategic improvement opportunities that could be operationally viable and lead to improvements in cost efficiency or quality. For Computer-Assisted Telephone Interview (CATI) surveys, research findings have indicated that the same data collection approach does not work effectively throughout an entire data collection cycle, stressing the need to develop a more flexible and efficient data collection strategy. To that extent, Statistics Canada has developed, implemented and tested a Responsive Collection Design (RCD) strategy on several CATI social surveys. RCD is an adaptive approach to survey data collection that uses information available prior to and during data collection to adjust the collection strategy for the remaining cases. In practice, the RCD approach monitors and analyses collection progress against a pre-determined set of indicators for two purposes: to identify critical data collection milestones that require significant changes to the collection approach and to adjust collection strategies to make the most efficient use of remaining available resources. In the RCD context, control of the data collection process is not determined solely by a desire to maximize the response rate or reduce costs. Numerous other considerations come into play when determining which aspects of data collection to adjust and how to adjust them. These include quality, productivity, the response propensity of in-progress cases, the collection mode and the competition from other surveys for collection resources. This paper presents Blaise implementation of the RCD strategy used for CATI social surveys. The highlights and lessons learned are described as along with the current and future RCD research plans and activities.

1. Introduction

Paradata research conducted over the past few years at Statistics Canada has indicated that the same collection approach does not work effectively throughout an entire data collection cycle. As Mohl and Laflamme (2007) have indicated, the data collection strategy used generally remains fairly static, i.e., a collection plan is developed prior to the collection start date using about the same collection approach from the beginning to the end of the collection period and specifying how collection effort (interviewer hours) will be applied. Once collection begins, collection plans are usually modified in response to the cumulative use of collection resources (proportion of budget spent) and progress. Therefore, operational paradata research has stressed the need to develop a more flexible and efficient data collection strategy for CATI social surveys, not only to maintain or reduce data collection costs but also to make better use of remaining available resources throughout the collection period. This approach implies an adaptive data collection or Responsive Collection Design (RCD) strategy. Responsive Design was first discussed by Groves and Heeringa (2006) for Computer-Assisted Personal Interview (CAPI) surveys. Mohl and Laflamme (2007) expanded the application of RCD to CATI surveys, developed an RCD conceptual framework and proposed several RCD strategies in the Statistics Canada context.

The framework proposed by Mohl and Laflamme (2007) includes two main components: active management (Hunter and Carboneau (2005) and Laflamme, Maydan and Miller. (2008a)) and adaptive collection. The main idea is to constantly assess the data collection process using the most recent paradata information available (active management), and adapt data collection strategies in order to make the most efficient use of available resources remaining (adaptive collection). In other words, RCD strategy aims to use information available prior to and during collection (accumulated paradata) to identify when changes to the collection approach are required in response to how well the collection progresses. The RCD strategy breaks down the survey data collection process into four different phases: planning, initial

collection, RCD Phase 1 (which aims to improve the response rate) and RCD Phase 2 (which aims to improve sample representativity).

The paper begins with an overview of the data collection context for CATI social surveys at Statistics Canada including a brief description of the main paradata sources available and those that are used for RCD. Section 3 presents the RCD strategy used for CATI surveys while Section 4 describes the tools, key indicators and approach used to actively manage the RCD surveys. Section 5 provides an overview of the highlights and the results obtained along with lessons learned. Finally, the current and future RCD research plans and activities are discussed in the last sections of the paper.

2. Data Collection for CATI Surveys at Statistics Canada

Data collection for CATI social surveys is conducted and managed in Statistics Canada's five Regional Office (RO) call centres located across the country. All survey applications are built using the Blaise software and the call scheduler automatically¹ assigns individual cases to interviewers working out of a centralized environment. The call scheduler takes into account the interviewers' profile, paradata information collected since the beginning of the data collection period (e.g., outcomes of the previous calls) and some data collection.

Interviewer profile

An interviewer profile is based on the interviewer's characteristics, skills and experience. It is an important component of the call scheduler. During the data collection period, a given interviewer can be identified to receive in priority (or exclusively) cases that belong to one primary interviewer group and one or more secondary interviewer groups. For example, experienced interviewers (or interviewers with very good convincing skills) are assigned to the Refusal group in order to try to convert those cases for which at least one refusal was recorded. The assignment of specific interviewers to the Tracing² group is also another very good example.

Collection parameters

In addition to the routing table, other collection parameters are considered by the call scheduler such as time slices, cap on calls, appointments and other technical parameters. The time slice feature in the CATI call scheduler was utilized to assist in managing the new cap on calls policy which limits the number of calls that can be made for each case. In practice, time slices ensure that a specific number of calls are attempted at different times of the day, and on different days of the week, before a case is finalized. It should be noted that only cases with a "no answer" treatment for the last call are subject to be influenced by the time slice parameters. The call scheduler also needs to manage appointments to make sure that cases are assigned at the appropriate time to an interviewer. In addition, some other technical parameters are considered by the call scheduler, for example, time between "busy" calls, minimum time between other "no answer", etc.

The management strategy for each survey can vary by regional office depending on the mix of surveys in collection and the workload and availability of interviewers. Survey management uses the standard Management Information System (MIS) and customized active management reports that are based on the Blaise Transaction History (BTH) files, Survey Operations Payroll System (SOPS) files for interviewers and sample design information available prior to data collection.

¹ Interviewers have the opportunity to use a browser tool to access any in-progress case. This means that the interviewer can scroll the list of all cases and manually select a case, thus skipping the call scheduler.

² Tracing consists of strategic and logical searches using all available resources to locate a respondent e.g. for those where the frame provided a wrong or missing telephone number.

Blaise Transaction History (BTH) record

A BTH record is automatically created each time a case is closed, whether it was opened for data collection or other purposes. The BTH record contains detailed information about each call made to contact each sampled unit during the data collection period. It also includes information on the survey and case identification, the date, the amount of time the case was open, the interviewer who worked on it, the resulting interviewer group (e.g., Refusal, Tracing, Regular, Home (finalized)), the result of the call (e.g., no contact, appointment, complete interview) plus additional relevant information. The call scheduler considers, for example, the number and time of calls that have been made to an individual case, the result and the interviewer group of the last call to assign cases to a given interviewer. These rules essentially refer to the ‘routing table’ of the survey application.

Survey Operations Pay System (SOPS)

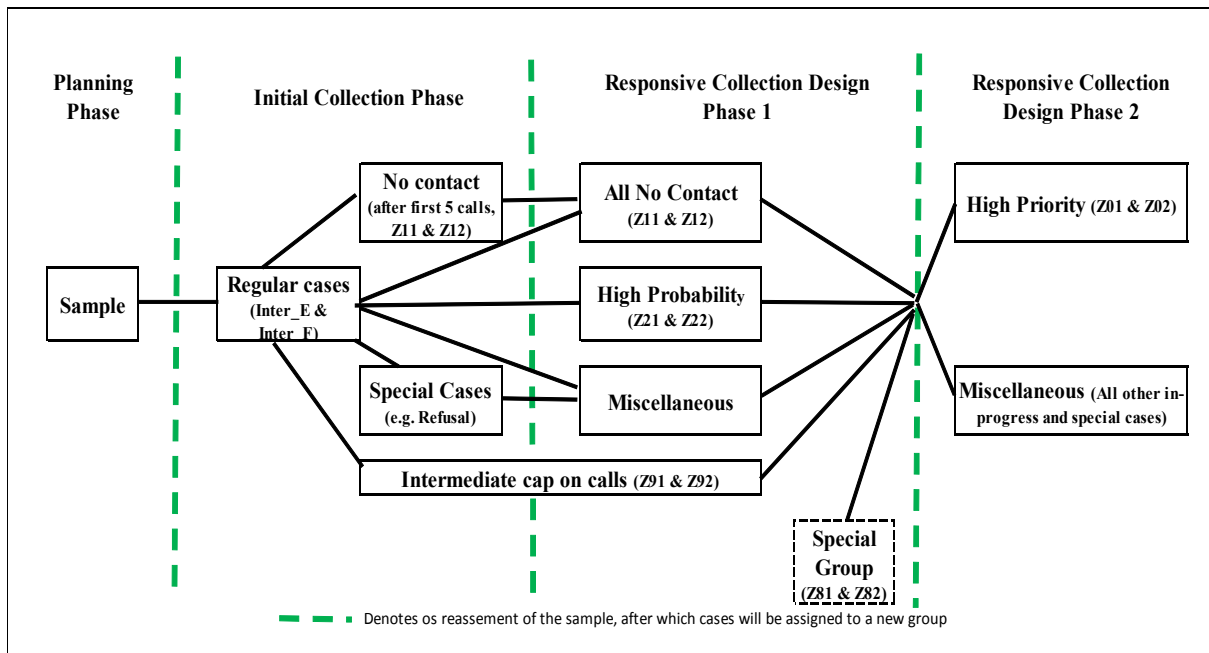
This file contains financial information about interviewer pay claims for all collection activities. A SOPS record is generated each time an interviewer enters a claim for a particular survey and task on a given day, either for direct data collection activities (interviewing, tracing, etc.) or for other purposes (supervision, specific training, etc.). Each claim includes the following: interviewer identification, type of interviewer (regular or senior), survey name, date, task code (interview, training, tracing, etc.), and number of payroll hours.

It is important to note that both BTH and SOPS paradata are accumulated throughout the collection period. The most recent information becomes available the day after a given transaction (call) took place or the day after an interviewer entered a claim. The timely availability and accessibility of this information are a key feature of the RCD approach.

3. Responsive Collection Design Strategy

Figure 1 presents a summary of the RCD strategy for Blaise social survey. The strategy is applied independently for every RO. The first phase (planning) occurs before data collection starts. During the planning phase, data collection activities and strategies are planned out, developed and tested for the other three data collection phases including the development of the propensity model(s). The second phase (initial collection) includes the first portion of the data collection process, from the collection start date up until it is determined that RCD Phase 1 needs to be initiated. An intermediate cap on calls was also introduced to avoid cases capping out before the last data collection phase. During this initial collection phase, many key indicators of the quality, productivity, cost and responding potential of in-progress cases are closely monitored to identify when the next RCD phase should be initiated. The third phase (RCD Phase 1) categorizes and prioritizes in-progress cases using information available prior to the beginning of collection and paradata information accumulated during collection with the objective of improving the overall response rates. During this phase, key indicators continue to be monitored. In particular, the sample representativity indicator provides information on the variability of response rates between domains of interest to help determining when the last phase should begin. The last phase (RCD Phase 2) aims at reducing the variance of response rates between the domains of interest (improving sample representativity) by targeting cases that belong to the domains with lower response rates.

Figure 1. RCD strategy for Blaise CATI surveys



3.1 Planning Phase

During the planning phase, data collection activities and strategies are planned and tested for the three following collection phases. In practice, RCD objectives, in-depth analysis of the previous collection cycle, sample validation, intermediate cap on calls, active management strategy and response propensity model(s) are investigated, developed and/or determined.

When applicable, the previous data collection cycles is analysed to validate the current sample, identify opportunities for improvement, develop a response propensity model to create high probability group(s) and determine collection strategies to be used in RCD phases. This analysis is also used to determine data collection parameters for the key indicators to identify critical data collection milestones for deciding when to move on to the next collection phase.

The concept of an intermediate cap on calls is used with two goals in mind. The first goal is to ensure that cases do not reach the global cap on calls (and then be resolved and sent to head office) too soon during the collection period. The second objective is to guarantee the best usage of the last few calls before cases reached the global cap by taking into account the characteristics and results of the previous calls.

A propensity logistic model is used to evaluate a household's likelihood of being interviewed during collection and to categorize and prioritize each in-progress case during RCD Phase 1. The surveys response propensity model(s) are developed for each regional office using different sources of information: sample design information, paradata available prior to the collection of the last collection cycle (when applicable) and paradata available and accumulated during the last collection cycle to identify the explanatory variables to be included in each model. In practice, sample design information (e.g., household composition and stratification variables), paradata from the previous data collection cycle available prior to collection (e.g., number of calls needed to complete previous interview, time of previous interview) and paradata from current collection cycle (i.e., variables accumulated since the beginning of the collection, such as the number of calls/contacts/appointments by period of the day, number of calls with specific outcome codes (e.g., refusal, tracing), number of calls after the first refusal or tracing outcome) are used in the propensity model to assign a response probability to each outstanding case in the sample. It should be noted that the variables included in the model(s) remain the same during the entire data collection period while the parameters of the model are re-evaluated daily using the most recent accumulated paradata at the end of each collection day. The model is developed and validated during the

planning stage for each RO to ensure that the model assigned higher completion probabilities to cases that ended up as completed interviews as opposed to cases that end-up as non-responses.

Active management refers to a set of plans and tools used to manage data collection while in progress. It is an essential and critical feature of any RCD strategy. At the planning stage, the four main steps of the active management approach (namely, survey monitoring, identifying/predicting problems (timely analysis), taking appropriate corrective actions and reporting/communicating) should be designed, developed and well tested. In particular, the reporting plan should identify the type of information needed as well as any variables that are required for reporting at various levels of aggregation and ensure that this information is available on a timely basis during the data collection period. The most important indicators for monitoring progress and which reports might be required at any given time during data collection (including the communication plan) also need to be determined during the planning phase since data needs vary depending on the point in time along the data collection process. The resulting active management information has also to be streamlined to avoid overwhelming survey managers. The active management strategy, tools and activities used in the following collection phases are described in more detail in Section 4.

3.2 Phase 2 – Initial Collection

Phase 2 (initial collection) is like regular CATI collection with the exception of one new: an intermediate cap on calls. Based on the analysis of the call patterns and their potential impact on the response rate, a cap on calls was introduced in 2006 for CATI surveys. This cap limits the maximum number of calls that can be made on one case in order to reduce respondent burden and make better use of available resources. Once a case reaches the determined threshold, the case goes into an intermediate cap group for further analysis. Detailed information about these cases is provided to each RO to make a decision on the best use of the last available calls for each case. It should be noted that “easier cases” (ones that required fewer calls and less effort to complete) are more likely to be contacted and interviewed during this phase.³ During the initial collection phase, key quality, productivity and cost indicators are closely monitored along with the responding potential of in-progress cases. These indicators are assessed separately for each RO to identify when to initiate RCD Phase 1 since their collection progressed at different paces.

3.3 Phase 3 – RCD Phase 1

RCD Phase 1 essentially aims at improving overall response rate. During that phase, paradata from previous survey available prior to collection or cycle and paradata accumulated during collection are used to assign a response probability to each outstanding case in the sample (by applying the response propensity model). Cases with higher scores have higher chances of being completed during the remaining data collection period. One of the main objectives is to use all relevant paradata information collected so far about each individual sample unit to reduce the amount of additional effort required to contact and get cooperation from remaining in-progress units or to identify a non-responding units. At the beginning of each day during the RCD Phase 1, all in-progress cases are categorized and prioritized based on the response probability and the analysis of the sequence of calls. Cases are grouped in the following way:

³ Previous research has shown that 40% to 50% of the total numbers of CATI respondents are reached on the first contact attempt (Laflamme (2008b)). Reaching the remaining half of respondents requires significant additional effort and a different collection strategy to achieve higher collection efficiency.

- The intermediate cap on calls group (created at the initial phase) continues to be used until the end of this phase. A case assigned to the intermediate cap group remains in that group unless it reached the cap on calls or is finalized.
- The ‘no contact’ group consists of all cases for which no contact with a respondent had ever been made since the beginning of the collection period (excluding cases in the intermediate cap on calls group). These cases are considered as either requiring an extra effort to make contact (hard-to-reach respondents) or as not having been given enough effort due to how the Blaise scheduler would bring up the cases. In either situation, separating these cases helps focus collection efforts.
- The high probability group consists of in-progress cases with the highest probability of completion as assigned by the propensity model (excluding cases in the intermediate cap on calls and ‘no contact’ groups). These cases are considered to be the most likely to be completed according all the information collected at some point in time during collection.
- The miscellaneous group consists of all other outstanding cases that are not assigned to one of the first three groups. In practice, these cases remains in the interviewer groups defined in the initial collection phase.

During RCD Phase 1, quality, productivity and cost indicators, the response propensity of the remaining cases continues to be closely monitored along with the representativity indicator to determine when a given RO should initiate the last RCD phase.

3.4 Phase 4 – RCD Phase 2

RCD Phase 2 aims at improving sample representativity by reducing the variance between the response rates among the domains of interest (e.g., sample design stratum). In other words, cases that belonged to underrepresented groups (i.e., groups with the lowest response rates) are prioritized during this last phase. In theory, the objective is to reach about the same targeted response rate in every domain of interest to improve sample representativity and to get more uniform non-response adjustment factors to reduce the variance.⁴ In practice, however, the expected and achieved response rates varied by RO (and by domain of interest within each RO). It is somewhat utopian to expect to obtain a 75% response rate in every sample design stratum because some individual regions and sub-regions are more reluctant to participate in surveys than others, recording historically and consistently lower response rates. During this last phase, sample representativity is monitored on a daily basis so that collection efforts could be switched between different domains depending on collection progress and the response rates obtained.

4. Active Management for Responsive Collection Design (RCD)

In general, active management is defined as a set of plans and tools to manage data collection while in progress. The purpose of the active management program is to provide timely, topical and relevant data on survey performance and progress throughout collection, so that problems during collection are identified early and decisions about how to correct problems are based on cost, effort, and data quality in addition to the response rates attained so far. Active management includes five different steps: planning, monitoring, identifying/predicting problems (timely analysis), taking appropriate corrective actions (when required) and communicating (including reporting).

⁴ It would be desirable to align the prioritization approach used during RCD Phase 2 with the non-response adjustment strategy used after collection (i.e., that both approaches should use about the same type of grouping (if applicable)).

First, active management requires planning at the time of application design and development. The plan should identify the paradata sources to be used, the most appropriate key progress indicators for that survey, the type of analyses to be performed, the types of actions that could be taken, and the strategy to deal with hard-to-reach cases. Planning could also include strategies for non-response follow-up as well as for moving cases from one mode to another. In sum, planning involves all aspects of data collection at all stages that are required to meet survey objectives. Second, monitoring survey progress is typically done through reports. The reporting plan for a survey should identify the type of information needed as well as any variables that are required for reporting at various levels of aggregation and ensure that this information is available on a timely basis during the data collection period. The plan should also identify which indicators are most important for monitoring progress and consequently which reports might be required for a given survey. Identifying and predicting problems represent the third step. The idea is to discover the problem when it occurs, or even before it happens, not when collection has ended. This requires a timely analysis plan and well-trained staff to quickly identify the nature of the problem. The monitoring reports do not presume to have all the answers to issues but are designed to identify possible problems and supplemental information that could guide or help the identification of the source of the problem. From that perspective, active management also provides mechanisms for handling unanticipated or emerging problems. This includes the ability to generate ad hoc reports to investigate situations as they develop. Taking appropriate corrective actions is the next step. The objective at this stage is to determine whether action is required to correct the problem and, if so, to decide what steps to take. The decision-making strategy and its implementation should have been planned at the initial planning stage. However, since the ability to take corrective action may depend on the design of the application and on operational constraints that might reduce the possible range of actions, it is important to include some contingency planning in the overall collection management plan. Finally, the communication plan has to identify the series of relevant reports that need to be distributed at different points in time during collection with the goal of not overburdening managers. The plan should clearly identify who needs to be contacted in the event of a problem, who has the authority to make decisions, and what the process should be for taking corrective action. The strategy should also identify who is responsible for monitoring different aspects of the collection process. A communication strategy that covers the collection phases of a survey can ensure that regional offices, methodologists and both survey collection managers and survey managers are informed of any decisions or discussions affecting data collection.

In the RCD context, active management has two objectives. The first one is to provide timely and relevant data on survey performance and progress based on key indicators and customized information, while the second one is to identify when data collection milestones are met and when changes to the collection strategy are required. The overall objective of active management is to make effective use of collection resources to find the most appropriate balance between data quality, timeliness and survey costs. The active management prototype tools used for the two RCD surveys consist of a series of SAS programs that have been consolidated into one master program. All paradata and other data sources are integrated and processed to produce a series of customized Excel reports on a daily basis.

The first two parts of this section describe the paradata sources and the key indicators used to actively manage both RCD surveys. The second part presents the strategy implemented to identify data collection milestones (i.e., when to initiate RCD phases) including other monitoring tools used to assess various aspects of data collection performance and progress (and to identify potential problems) at different points in time during data collection.

4.1 Paradata and Other Sources

Active management for RCD uses numerous data sources (i.e., paradata and other data sources) to monitor data collection performance and progress, namely, Blaise Transaction History (BTH) files,

Survey Operations Payroll System (SOPS) files for interviewers, budget information, sample design and targeted response rates. The information can be available prior to data collection (e.g., sample design, budget information) or can become available at the beginning of collection and accumulated throughout the collection period (e.g., BTH and SOPS). In practice, the most recent BTH and SOPS information becomes available the day after a given transaction (call) takes place or the day after an interviewer enters a claim in the case of SOPS. The timely availability and accessibility of this information are key features of the RCD strategy.

Many other sources of information are also used. Budgeted system time for production and interview claim hours figures planned at the beginning of the survey are used to determine the proportion of resources still available at the end of each collection day. The sample design and frame information (e.g., demographic characteristics of sample units and stratification) are used in conjunction with the paradata to enhance the analytical value of active management monitoring reports. Finally, it should be noted that the results of the response propensity model that uses data from previous and current collection cycles for each of the two surveys are also used indirectly in the active management strategy.

4.2 Key Indicators

Response rate is not the only measure that should be used to monitor and assess data collection performance and progress. Instead, response rate should be used in conjunction with other measures such as survey productivity, cost and representativity indicators to make the best use of data collection resources while taking into account the trade-off between quality, timeliness and cost. Therefore, the following key indicators have been developed and implemented to better evaluate and monitor progress, quality and performance during data collection in the RCD context and to make more informed decisions about changes to the data collection strategy. The indicators are presented in a practical application in Figure 2 in section 4.3.

The productivity indicator is defined as the ratio of the system time devoted to the interviews themselves to the total system time including all unsuccessful and successful calls (Laflamme, 2009). Total system time represents the total time logged onto the system once a case is open. For RCD active management, two productivity indicators are defined: daily productivity and average productivity over the last five days. The purpose of the average productivity indicator is to smooth the daily series to reduce the daily variability that is more likely to occur during the weekend or at the end of the collection period.

The responding potential of in-progress cases is based on two measures. The first one represents the proportion of regular in-progress cases while the second one provides an indication of the effort already put into these cases as a function of the global cap on calls of the survey. The regular in-progress cases refer to cases (with at least one call made) that are not resolved (finalized) and for which no previous outcome(s) resulted in a refusal, tracing or special outcome. This includes cases with incomplete response, appointments, no contact or outcome codes other than the three mentioned above (refusal, tracing or special outcome). The proportion of regular cases in Figure 2 is simply the number of regular in-progress cases divided by the total sample size. As shown in Figure 2, the proportion of regular in-progress cases reaches a peak on the 6th day of collection where all cases were called at least once and about 30% of the sample units were resolved. After this day, the proportion decreases continually because cases are likely to change from their regular status. A relative measure of the effort already put on these regular cases can be obtained by taking the ratio of the average number of calls made for the regular in-progress case divided by the global cap on calls for the survey.

The proportion of the budgeted interviewer payroll and system time hours spent since the beginning of the survey are both used as cost indicators. They are defined respectively as the cumulative interviewer claims (hours spent) divided by budgeted interviewer claims in hours and the cumulative

system time spent divided by budgeted system time. These two indicators are generally well aligned (i.e., they evolve at the same pace) during data collection. Finally, the last indicator showed in Figure 2 is the response rate, which represents simply the cumulative response rate since the beginning of the survey. The series of key indicators is used to monitor and evaluate data collection performance and progress and also to determine data collection milestones where changes to collection strategy are required (i.e., when RCD Phase 1 and Phase 2 should be initiated). These milestones are first identified during planning and then fine tuned, if necessary, during collection.

The representativity indicator is defined as (1 minus the standard deviation of response rates between the domains of interest). The domains of interest are based on the sample design information available prior to the beginning of the collection period. The implemented representativity indicator for RCD is conceptually different than the R-indicator proposed by Schouten, Cobben and Bethlehem (2009). For example, no assumption is made about the response propensity of each sample unit prior to collection. The objective was to develop a simple, standard and practical representativity indicator for data collection monitoring and operational purposes that can be easily implemented for all surveys.

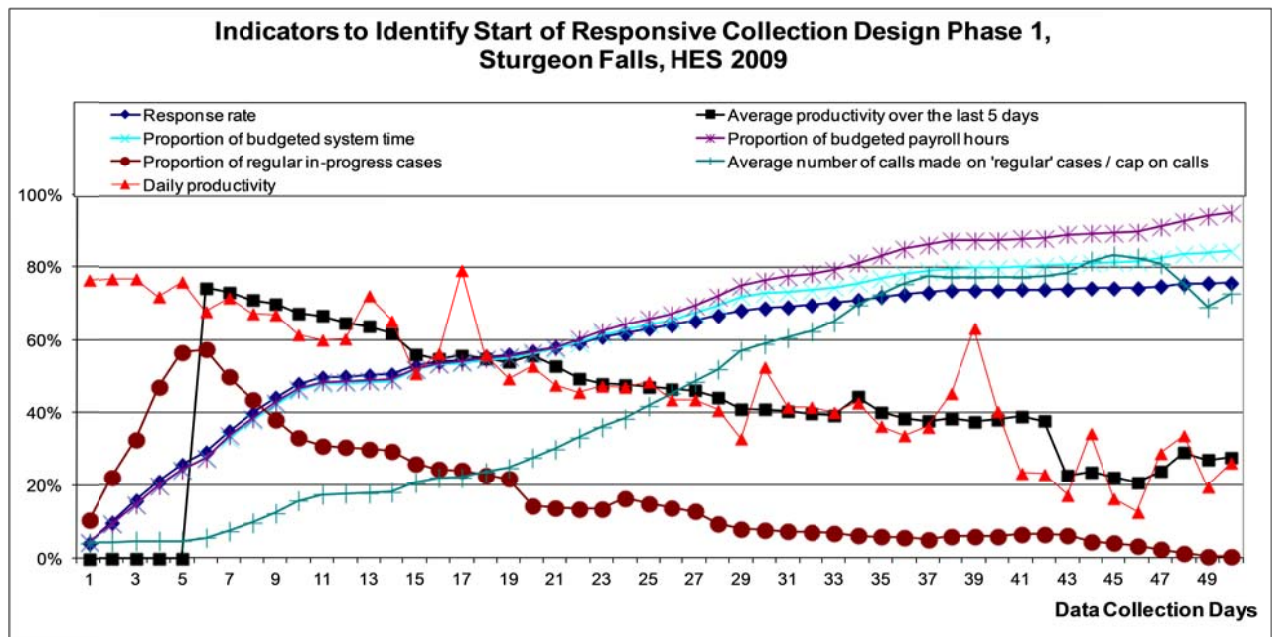
4.3 Determining Critical Data Collection Milestones

The decision process used to determine when to move from one data collection phase to another one needs to be planned prior to the beginning of data collection. Decisions about collection strategy changes for RCD are based on cost, productivity, the responding potential of in-progress cases and data quality indicators in addition to response rates. During the planning phase, information from the previous data collection cycle is used to re-calculate all key indicators for each survey by Regional Office.⁵ One of the main objectives of this analysis is to determine the conditions or thresholds to be used for each key indicator to identify when changes to the collection strategy are required. Once decided, the same thresholds is used for all RO's even though the monitoring (and decision process) is done independently.

As shown in Figure 2, the response rate increases at the same rate as costs (% of budget spent expressed as % of budgeted system time and % of budgeted payroll hours) at the beginning of the survey. However, once average productivity over the last 5 days reached about 45% (around the 26th collection day), the gap between response rate and the two cost indicators started growing (the same effort produced fewer interviews than at the beginning). Early on, the proportion of in-progress regular cases started to decrease while the ratio of the average number of calls made for the regular in-progress cases divided by the global cap on calls for the survey continued to increase rapidly, also suggesting that effort was being spent on a smaller number of cases with less productivity. This is particularly obvious after the lines of these last two series crossed on 18th day.

Figure 2. Key indicators

⁵ It is worth noting that the shape of the series in Figure 2 and their relationship are very similar by RO even though they tend to progress at a different pace. This figure represents the RCD pilot collection process for HES 2009 in the Sturgeon Falls office. It is used as an example to illustrate the thought process used in the planning phase.



For HES 2009, we used a conservative⁶ approach, the window to initiate RCD Phase 1 was identified to be between the 26th and 32nd day of data collection for this particular RO⁷. This critical window was defined using the following six conditions based on key indicators for each RO summarized in Table 1. This exercise was repeated for each RO.

Table 1. Conditions for RCD Phase 1 and Phase 2

Condition Number	Type of indicator	Conditions description	RCD Phase 1	RCD Phase 2
1	Quality	Response rate	> 50%	> 60%
2	Quality	Average response rate increases over the last 5 days	n/a	< 1%
3	Budget(cost)	% of budgeted payroll hours	> 50%	> 80%
4	Budget(cost)	% of budgeted system time	> 55%	> 80%
5	Productivity*	Initial minus current productivity*	> 15 pp**	> 25 pp**
6	Responding potential of in-progress cases	% of 'regular' in-progress cases	< 25%	< 20%
7	Responding potential of in-progress cases	Average number of calls made for the 'regular in-progress' case divided by the global cap on calls	> 32%	> 64%

*Productivity is the average productivity over last 5 days

**pp means percentage points

During the initial data collection phase, all these key indicators are closely monitored throughout the collection period for each RO on a daily basis. In order to facilitate interpretation and decision making for survey managers, the indicators and the status of each condition at a given point in time during collection are integrated and consolidated into one single dashboard as shown in Table 2. The value of each indicator and condition (1 means that the condition is met) is presented in the dashboard. The sum of the conditions is used to determine the status of each RO with respect to the initialization of the next phase. Therefore, when the sum of the 6 conditions is between 1 and 3, there is no need to

⁶ For current RCD, we start the RCD Phase 1 earlier to take more advantage of the responsive design approach.

⁷ The decision process used to determine when to move to the next data collection phase also took into account operational constraints such as weekend and Labour Force Survey (LFS) week as well as advices from RO.

start the next phase. However, when the sum of conditions equals 4 (yellow line), it indicates that this RO is approaching the threshold for moving to the next phase. Finally, when the sum of conditions adds up to 5 or 6 (red line), the RO should switch to the next phase if it has not already done so. The colours provide an easy way to evaluate the current status of each RO. It should be noted at this point that further research is required in order to more objectively identify the optimal data collection milestones where changes to data collection strategy are required.

Table 2. Dashboard of key indicators for RCD Phase 1

RO	Response Rate		Productivity (Average over last 5 days)			Budget and Cost				Propensity of In-Progress Cases				Sum of conditions
	Response Rate	Cond 1	Initial	Current	Cond 2	% of budget payroll hours	Cond 3	% of budget system time	Cond 4	% of regular in-progress cases	Cond 5	Avg. # of calls for regular in-progress cases / cap	Cond 6	
EDM	52.2%	1	64.3%	41.6%	1	56.6%	1	62.4%	1	28.3%	0	7.2	0	4
HFX	52.6%	1	63.6%	45.5%	1	62.8%	1	69.8%	1	27.0%	0	7.7	0	4
SHER	46.3%	0	66.9%	58.3%	1	49.8%	0	50.7%	0	34.5%	0	3.4	0	1
STURG	64.2%	1	74.4%	46.6%	1	67.2%	1	65.6%	1	13.9%	1	11.3	1	6
TOR	52.6%	1	65.7%	53.6%	1	58.0%	1	55.4%	1	27.9%	0	6.8	0	4
WIN	62.4%	1	69.9%	46.8%	1	69.8%	1	69.0%	1	12.5%	1	13.0	1	6

4.3.1 Moving from RCD Phase 1 to RCD Phase 2

During RCD Phase 1, the same key indicators used in the initial phase are monitored along with two additional ones (the representativity indicator⁸ and the average response rate increases over the last 5 days) to determine when a given RO should initiate RCD Phase 2.

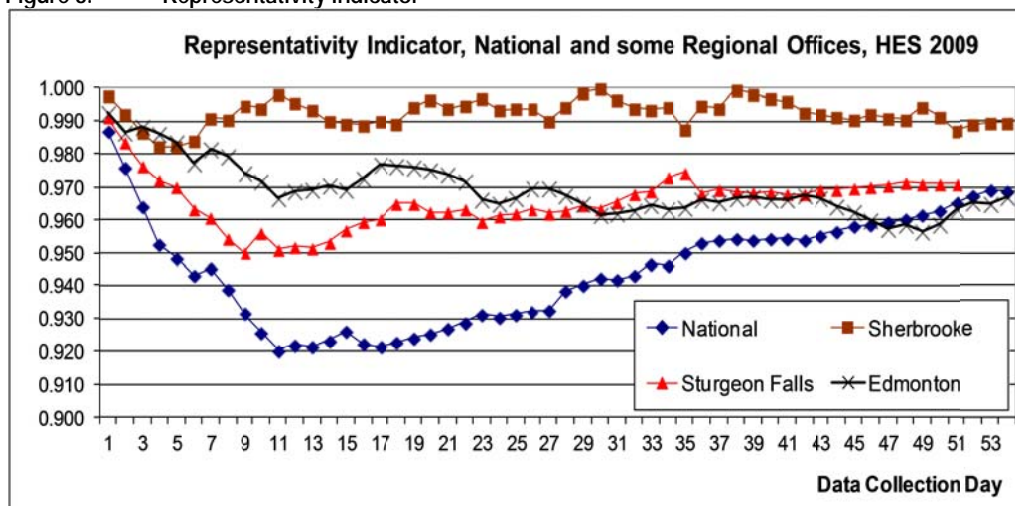
The decision to initiate the last phase is based on these key indicators with a new set of conditions (see Table 1). Another dashboard (similar to Table 2) is also produced to monitor collection during RCD Phase 1. The representativity indicator is only used as a qualitative measure to evaluate the trend of sample representativity over time, that is, no specific conditions is set for this indicator during the planning phase. This last phase only prioritizes cases (i.e., no sub-sampling) that belong to under-represented groups of interest (groups with the lowest response rates)⁹. It is important to note that the RCD Phase 2 should not be initiated (if required) too late during data collection to provide some flexibility and time to improve sample representativity. The representativity indicator provides a summary measure within each RO as well as at the national level. This approach could potentially result in “conflicting” objectives. For example, the representativity indicator could be high (close to 1) in one particular RO while its overall response rate is lower than the national response rate¹⁰ (see Figure 3). In other words, while no group priority would be required at the RO level, response rates would have to be increased in all domains to improve the national representativity indicator. In practice, the expected and achieved response rates varied by RO, and these have to be taken into account to set realistic regional and national response rate expectations.

⁸ The representativity indicator defined as (1 minus the standard deviation of response rates), tracking the variance between response rates in domains of interests (see Figure 3)).

⁹ It should be noted that cases are still subject to the cap on calls policy during this last collection phase and this also have to be considered during implementation because prioritized cases are more likely to reach the global limit of calls.

¹⁰ The national representativity indicator is lower than the regional indicators because ROs generally progress at different paces and the geographical dimension is often considered in the domains of interest.

Figure 3. Representativity indicator



4.4 Other Monitoring Tools Used

The analysis of the previous data collection cycle during the planning phase is also used to improve existing active management tools and develop new approaches to closely monitor collection progress, to assess data collection performance and to identify/predict problems during data collection. The adopted strategy takes advantage of the active management experiences with many previous surveys. This section provides an overview of some of the most important tools used during collection.

4.4.1 Data Collection effort and Progress

Data collection effort, progress and results are monitored by RO. A comprehensive report provides daily and cumulative totals for all key indicators and other relevant measures since the beginning of the data collection period. In practice, this report consolidates the most important cumulative indicators shown in Figure 2 (including their sub-components) and a series of other measures that provide complementary information to get a better appreciation of the operational context at any point in time during collection. The report does not only give a snapshot of current survey progress status, but also provides the daily effort, progress and results since the beginning of data collection. For example, the survey progress can be monitored using the daily and cumulative number of resolved cases (and consequently the proportion of in-progress cases) at the end of each data collection day by final status of the cases (respondents, non-respondents, out-of-scope and capped cases). The achieved response rate and number of respondents can then be compared with the expected targets at any point in time according to the planned and expected survey progress. The no contact, refusal and tracing rates as well as the refusal rates at the first contact are other examples. The rest of this section presents in more detail some of the important active management tools used during collection of RCD surveys.

4.4.2 In-Progress Cases

During collection a lot of emphasis is spent on the in-progress cases to monitor the effort already put on these cases, to evaluate their characteristics and to assess their relative likelihood of being interviewed. At any point in time during collection, changes to the collection strategy or corrective actions will only impact in-progress cases: no action can be undertaken on resolved cases except in special circumstances (such as non-response follow-up, re-interview).

Distribution of the number of calls for in-progress cases

The effort already spent on the in-progress cases is monitored by the distribution of the number of calls for these cases (but it could have been done by the system time spent by type of in-progress case). At first, this distribution is particularly useful to assess and predict the proportion of cases that will be sent to the intermediate cap group during the initial phase and RCD Phase 1. Later, this distribution is used to evaluate and predict the number of cases that will possibly reach the global cap on calls or be resolved. For example, it is not sufficient to monitor the proportion of cases that reach the global cap on calls to actively manage collection because the proportion of cases that will eventually reach this cap (and then be resolved if no action is taken) also needs to be monitored. This report can also be used to identify the proportion of cases with a low number of calls at the end of the RCD Phase 1 that might require specific attention through the special group as seen in Figure 1.

Distribution of in-progress cases by interviewer group

During collection, cases can be assigned to different interviewer queues (Inter (regular cases), Refusal, Tracing, Home (resolved), etc.) depending on the outcome of the calls. At the start of collection, cases are generally assigned to the regular case group. As the survey progresses, some cases are resolved, some are recorded as refusals and some others require tracing and are redirected to the interviewer queue. It means that the characteristics and portfolio of in-progress cases change during collection. In fact, as the survey progresses, the overall responding potential of in-progress cases decreases (along with survey productivity) because cases are on average less likely to respond. Therefore, the distribution of in-progress cases by interviewer group as well as the average number of calls made for cases in each group is monitored throughout collection. Since different groups are used in the data collection phases, there is also a need to provide each RO with guidelines to help with interviewer staffing, especially at the beginning of the RCD Phase 1.

Expected distribution by interviewer group between phases

A practical challenge ROs are facing is balancing a number of interviewers assigned to work on each of these groups. On the one hand, if too many interviewers are assigned to a group, there is the possibility that they would run out of cases. On the other hand, if not enough staff is assigned to a particular group, some cases would not be tried. The objective of this tool is to foresee the interviewing effort that will be required for each group in advance of starting RCD Phase 1. In other words, by the end of the initial collection phase, the distribution of in-progress cases among the groups for RCD Phase 1 (no contact, high response probability groups, etc.) is simulated (based on the current status of the cases) to provide each RO with guidelines to help with interviewer staffing at the beginning of RCD Phase 1. In practice, the report gives the distribution of cases that would be observed if RCD Phase 1 had started on this day. A second distribution is also provided, adjusting the distribution of cases to take into account the expected productivity of each group in RCD Phase 1. For example, even though the 'High probability' group represents less than 5% of the in-progress cases at the end of the initial collection phase, it does not mean that only 5% of the working force (interviewer staffing) needs to be assigned to this group because its expected productivity is more likely to be higher than other groups (e.g., the refusal group). The percentage of effort suggested is then derived by adjusting the proportion of in-progress cases by the expected productivity of each group in RCD Phase 1. In summary, this special report is essentially produced to provide ROs with guidelines for next-day planning of interviewers' assignments (data collection effort) between the various groups. These guidelines have to be adapted for each Regional Office according to their operational constraints (monthly planning of staff by some) and capacity (limited number of tracing or refusal interviewers). This information is very important prior to the initiation of a new phase in order for regional managers to plan and maximize the allocation of interviewers.

4.4.3 Special Data Collection Activities

The purpose of this monitoring tool is to assess and monitor effort and progress for special data collection activities which include four particular types of cases: cases with at least one refusal outcome, cases with at least one tracing outcome, cases redirected at least once to Senior Interviewer (SI) or Project Manager (PM) Blaise groups and cases redirected to the intermediate cap on calls group.¹¹ For each of these four types of cases, the proportion of these cases still in-progress, the average number of calls made on both resolved and in-progress cases as well as the number (and the proportion) of cases converted to an interview are monitored. In addition, the report gives the distribution of the total number of calls made after the first occurrence of each type of case (e.g., after the first refusal). This can be seen as a measure of effort made by RO on the hard-to-reach and hard-to-interview cases.¹² These reports provide a comprehensive tool to assess the responding potential of the overall in-progress cases, particularly when close to the end of the data collection period. For example, in HES 2009, about 15% of the initial sample was still in-progress during the last few days of collection. By analysing the characteristics of these in-progress cases, it can be seen that the proportion of regular in-progress cases is very small (see Figure 2). Most of these in-progress cases are attached to one (or more) of the four types of particular cases for which a lot of effort had already been spent. In other words, the marginal daily gain in terms of response rates is generally very small close to the end of collection because the responding potential of the overall in-progress sample decreases as the survey progresses.

4.4.4 Survey Progress by Domain of Interest and Sample Representativity

This report provides a detailed view of survey progress in terms of response rate by domain of interest by taking advantage of sample design information available at the beginning of the data collection period. In particular, it compares the expected response rates (or number of respondents) with those actually achieved at a level of aggregation not available in the standard survey reports. Used in conjunction with the representativity indicator, it also provides a tool to prioritize domains of interest in RCD Phase 2 for which progress is below expectations.

4.4.5 Ad Hoc Reports

The active management reports do not aim at giving answers to all possible issues. They are planned and designed from past experiences to identify possible and common problems along with other data or information that could explain the source of these problems. However, they are also built in such a way as to highlight key elements that could either identify the source of the emerging or new issues, or point to the need for more in-depth investigation through the use of ad hoc investigations. These reports can be produced in a timely manner as they take advantage of the consolidated information produced by the active management infrastructure.

¹¹ A case can fall into more than one of these types of cases.

¹²Refusal cases are of particular concern since Statistics Canada limits the number of follow-up contacts with a respondent once he/she states their unwillingness to cooperate.

5. Highlights of RCD Pilots

This section summarizes the results obtained as well as the highlights from the first two Responsive Collection Design surveys. In particular, response rates, data collection effort, productivity and representativity as well as the impact of the categorization and prioritization of cases in RCD Phase 1 and Phase 2 are discussed.

5.1 Response Rates, Data Collection Effort, Productivity and Representativity

Table 3 indicates that both HES 2009 and SLID 2010 achieved a better response rate under RCD than in their previous collection cycle (even with a higher proportion of difficult cases for both surveys).

Table 3. Response rates and representativity indicators for the previous two cycles of HES and SLID

	HES		SLID	
	Response Rate (%)	Representativity Indicator	Response Rate** (%)	Representativity Indicator
Previous survey (HES 2007, SLID 2009)	72.60	0.9593	71.10	0.8279
Last survey* (HES 2009, SLID 2010)	74.10	0.9704	72.20	0.8213

*RCD was only used in the last survey cycle for both HES and SLID

**For SLID, the overall response rate uses generated cases (included in the Control group) which had a relatively poor response rate. These cases are excluded for all RCD and Control group comparisons

When comparing within each survey the response rate between the RCD and control groups (Table 4), the differences are very small.

Table 4. Response rates, effort and productivity per data collection phases for HES and SLID

	HES - 2009			SLID - 2010		
	Response Rate (%)	Effort (System Time - hours)	Productivity (%)	Response Rate (%)	Effort (System Time - hours)	Productivity (%)
Control Group (CG)	74.0	4,841	53.1	73.0	13,133	35.1
Initial	57.2	3,346	60.6	55.8	5,405	67.4
RCD phase 1	14.3	1,262	37.5	26.0	3,021	23.7
RCD phase 2	2.2	233	30.3	19.2	4,707	5.4
Responsive Collection Design (RD) group	74.1	4,743	54.2	72.8	12,875	35.5
Initial	56.1	3,253	61.3	55.8	5,320	67.5
RCD phase 1	14.7	1,218	40.1	22.7	2,280	27.6
RCD phase 2	2.9	272	33.3	21.7	5,274	6.5

However, about the same response rates were achieved with less collection effort (calls and system time) for the RCD group than for the control group. For example, for HES, the average numbers of calls per case for the RCD and control groups were 7.3 and 7.6, respectively. For SLID, the corresponding numbers were 10.6 and 10.8. Furthermore, for both surveys, about 2% less system time was used for the RCD group than for the control group.

Table 4 shows response rates, data collection effort and productivity for the two surveys by data collection phase. Survey productivity is defined as the ratio of the amount of system time devoted to actual interviewing to the total system time (which includes both successful and unsuccessful calls). For HES, productivity was higher for all phases with less effort spent on the first two phases (initial and RCD Phase 1). For example, about 3% less system time was allocated to the initial phase for the RCD group. This could explain why the response rate for RCD group is slightly lower than that obtained by the control group and not equal as one would expect. SLID productivity was also higher

for all phases of the RCD group. However, contrary to our expectations, only RCD Phase 2 showed a better response rate. In RCD Phase 1, the RCD group had a lower response rate (22.7%) than the CG (26.0%).

The difference in rates is partially due to the fact that less effort was expended on the RCD group (24.5% less system time) during that phase. Productivity was only marginally higher for the RCD group during the initial phase of collection, but a wider gap appeared in RCD Phases 1 and 2. This occurred for both surveys. Higher productivity was expected for the two RCD since less time was spent on RCD groups (about 2% less compared to the control group) and both achieved similar response rates.

Table 5. Some statistics for in-progress cases at the end of collection for HES and SLID

	HES - 2009		SLID - 2010	
	CG	RCD group	CG	RCD group
End of collection - In-progress cases				
Percentage of cases	8.5%	13.3%	5.5%	6.3%
Average number of calls	12.7	15.2	18.0	20.0
Percentage of cases with 5 calls or less	24.0%	13.1%	7.6%	5.4%
End of collection - Capped cases (40 calls)	6.9%	2.9%	5.9%	4.9%

Finally, for HES cases remaining in-progress at the end of collection, about 15.2 calls were made on average for RCD cases compared to 12.7 for CG cases as shown in Table 5. In particular, at the end of collection, about 24% of in-progress cases in the CG had 5 calls or less compared to 13% for the RCD group. This observation again suggests that the effort was applied more effectively for the RCD group while leading to the same response rate. Overall, the variability of the distribution of the number of calls was smaller for the RCD group. In other words, the CG had more cases with few attempts (5 calls or less) as well as more capped out calls (especially for HES). Similar results were obtained for SLID but with smaller differences between the two groups. It might be possible that survey managers learned well after the first RCD survey.

5.2 Categorization and Prioritization of Cases in RCD Phase 1 and Phase 2

The paradata information collected throughout the collection period (including the sequence of calls) along with the results of the response propensity models permit the categorisation of each in-progress case in practical, meaningful and more homogeneous groups. This approach also facilitates continuing collection by assigning the grouping of cases to interviewers through the Blaise call scheduler according to interviewers' skills and profiles during data collection.

5.2.1 Global and Intermediate Cap on Calls

The idea of an intermediate cap on calls in the RCD context is to set aside cases in the RCD group that reached this threshold during the initial phase or RCD Phase 1, so that the best use of the remaining calls can be made. For HES, the number of cases that reached the intermediate (20) and global (25) caps in the RCD group were 10.5% and 2.9%, respectively. In comparison with the CG, 12% of cases were called 20 times or more and 6.9% of cases reached the global cap on calls. It is clear that less collection effort (both in terms of average of calls (22.5 versus 23.8) and system time (163 hours versus 86 hours) was devoted on average for cases with 20 calls or more in the RCD group compared to the CG, which impacted the response rate. The response rate for the cases with 20 calls or more were 14.1% and 17.8% of the RCD group and CG, respectively. However, it should be noted that the

productivity was higher for the RCD group (36.7%) compared to 31.5% for the CG, suggesting that the collection effort was more efficiently used in the RCD group. However, in future RCD, effort will need to be better monitored in real time in order for capped cases to progress during collection and not just at the end. Because of technical problems with SLID, the analysis of the impact of the intermediate cap on calls was compromised, and there are no findings to report.

5.2.2 High Response Probability Groups

The RCD groups of both surveys experienced higher response rates at the end of RCD Phase 1 for cases that were placed at least once in the high probability group compared to similar cases in the RCD group (i.e., other regular cases not identified as high probability units in RCD Phase 1).

Table 6. Response rates for high probability group by RCD phase

		Response rates			
		HES - 2009		SLID - 2010	
Phase	Type of Cases	CG	RCD group	CG	RCD group
End of RCD Phase 1	Cases identified at least once in high probability group	n/a	44.4%	n/a	37.8%
	Cases identified as regular, but not in high probability group	39.0%	33.2%	33.9%	26.2%
End of RCD Phase 2 (End of Collection)	Cases identified at least once in high probability group	n/a	0.0%	n/a	0.0%
	Cases identified as regular, but not in high probability group	55.1%	0.0%	58.1%	0.0%

During the same period, the response rate for the CG for the same type of cases was lower than for the high probability RCD group for both surveys. In other words, the propensity model has properly identified cases more likely to respond and placed them rightfully in the high probability group. At the end of collection, the response rates for cases identified at least once for the high probability group were still higher than the rates obtained for the same type of cases in the control group. For SLID, a high probability tracing group was also created. Its response rate was 14.6% compared with 10.8% for tracing cases from the control group at the end of RCD Phase 1. Again, the propensity models seem to have helped pick out the cases that were most likely to complete the survey.

5.2.3 Non-Contact Group

The final response rates for cases that did not have any contact made at the beginning of RCD Phase 1 were 36.2% for RCD and 34.2% for CG for HES, and 14% for the RCD group and 15.8% for the CG for SLID.¹³ It appears that the overall benefit of separating non-contact cases into a different group was modest. However, the prior knowledge of the cases' status provides added-value to interviewers (e.g., these cases have generally a lower productivity). Further experiments and research are suggested to allow more definite conclusions about the characteristics of respondents and the most effective collection methods for these cases.

¹³ The RCD non-contact group for SLID received 12% less effort in terms of system time than the non-contact cases in the control group.

5.3 Responsive Collection Design Phase 2

While the objective of RCD Phase 1 is to increase the overall response rate, the goal of RCD Phase 2 is to improve the representativity of the overall sample (not only for the RCD group) by reducing the variance in response rates between the domains of interest since the RCD groups and CG are combined during RCD Phase 2. Targeted cases that belong to domains of interest are put in priority groups during RCD Phase 2. However, no sub-sampling of cases is done, i.e., all in-progress cases can be worked on during this last phase.

Regarding the representativity indicator (or homogeneity of response rates), there was a gain for HES 2009 compared to HES 2007 (Table 3). In other words, there was less variability among response rates in domains of interest in 2009. To achieve this, during the RCD Phase 2, about 16% more system time was devoted to the RCD group than the control group. For SLID, the results are not as clear cut: there was an improvement in several regional offices, but representativity deteriorated somewhat at the national level. It is possible that some features of a longitudinal survey, such as differential attrition rates, may make it more difficult to maintain representativity over time. Further work using SLID data is needed to test this conjecture. Finally, RCD Phase 2 should not be initiated too late during data collection in order to provide some flexibility and time for improving sample representativity.

5.4 Analysis of RCD Variances

Studies were conducted for HES and SLID in order to evaluate the impact of the RCD on the estimates and on the variance estimates. The weighting process of both surveys took into account the fact that two different collection strategies were used, especially during RCD Phase 1. In other words, the non-response adjustment was adapted to consider the split sample for both surveys.

5.4.1 HES

The weighting strategy for HES can be briefly described as follows: 1) Derive initial HES weights based on the final CCHS weights; 2) Adjust for HES non-response and; 3) Obtain final weights by calibration. The non-response adjustment for the HES process was performed separately for the two groups (RCD and Control groups). Out of the 198 estimates, only 10 estimates were statistically different between the control group and the RCD group (using a 95% confidence level). These results suggest that no significant bias was introduced by the use of the RCD. Similarly, the impact of the RCD on the variance estimates was studied. It was found that the differences between the CVs of the RCD group domain estimates and those of the control group domain estimates were generally small and not systematically in the same direction. In addition, another study was done using the Kish statistic. This statistic only uses the weights; it does not require the selection of a set of variables. A number of alternative weighting strategies were studied and it was again concluded that both the control group and the RCD group were leading to similar variance estimates. In conclusion, the investigations for HES have shown that the domain estimates and the variance estimates produced using the units in the RCD group were not significantly different from those produced using the units in the control group.

5.4.2 SLID

Although SLID's weighting strategy is very complex, it can be summarized in four distinct steps: 1) Derive initial SLID weights based on the adjusted non-response Labour Force Survey (LFS) weights;

2) Adjust for SLID non-response; 3) Carry out other adjustments (e.g., Influential, Migration) and finally; 4) Obtain final weights by calibration.

As for HES, the non-response adjustment model for SLID (step 2) was modified to include an indicator for RCD and nine paradata variables to account for the RCD. First, two sets of non-response weights were generated: one using the old strategy (LFS variables only) and one in which paradata variables (including the RCD indicator) were combined with LFS variables. Both sets of weights then underwent other adjustments (step 3) and calibration (step 4) using the same set of auxiliary variables. Estimates and variance estimates were then produced for both sets of calibrated weights for selected SLID variables. In general, both methods yielded estimates that were statistically equivalent at all levels for all variables studied with very few exceptions. There was a slight tendency for the RCD model to yield slightly higher variance estimates (though not systematically). Secondly, units collected through RCD and units collected through regular SLID collection procedures were treated as two independent samples. Non-response adjusted weights were produced for each of the following scenarios: 1) Control sample using LFS variables only in the non-response adjustment; 2) RCD sample using LFS + Paradata variables in the non-response adjustment and; 3) RCD sample using LFS variables only in the non-response adjustment. For each set of non-response adjusted weights, calibrated weights were produced and, in turn, estimates and variance estimates were derived. Comparisons were made between two sets of estimates through statistical tests at the 5% level. In general, although there were some differences between the estimates produced using different non-response adjustment methods, these differences were not large enough to be statistically significant in the vast majority of cases. While the RCD method yielded point estimates similar to the regular process, it had a tendency to yield slightly higher estimates of variance. Work is being carried out to fully understand why there was an increase in the variance.

6. Lessons Learned

This section provides an overview of the lessons learned from these first two Responsive Collection Design surveys. The RCD feasibility and complexity, the active management, the communication, the staff knowledge, some technical aspects and interviewer staffing levels will be discussed.

6.1 RCD feasibility and complexity

Globally, the RCD concept for CATI social surveys is proved and viable. The RCD strategy used for SLID was slightly modified to take advantage of the lessons learned in the first RCD and to respond to its specific needs. It should be noted that SLID is one of Statistics Canada's most complex CATI surveys and yet it was still possible to use RCD effectively. For example, the prioritization and categorization of cases under the RCD (e.g., high probability and high probability - tracing during RCD Phase 1) had a positive impact on response rates, i.e., the propensity models had properly identified cases that were more likely to respond.

From a conceptual standpoint, it is essential to go through several thought processes during the planning phase before implementing an RCD. For example, when paradata from previous data collection cycles (or paradata for similar surveys) are available, simulations can be used to test the strategy and RCD active management programs. The different thresholds required to determine when to initiate RCD phases can be identified. It is also necessary to identify all the specificities of the survey, to incorporate them into the process if necessary and to thoroughly test all the modifications with their impact on the process. As we speak, the RCD surveys sample is no longer be split into two groups (RCD and CG). This eases data collection management in regional offices. However, this approach does not prevent the implementation and analysis of embedded experiments.

6.2 Active management

In a RCD context, active management has two main objectives: to provide timely information on survey progress and performance and to decide when is the right moment to initiate RCD phases during collection. The pilots clearly demonstrated that the active management strategy in place was able to meet these two objectives and even more. However, these two surveys also revealed some issues for future RCD surveys. For example, one of the main challenges for active management is to produce relevant and manageable reports that can be easily used during collection and to avoid overwhelming survey managers with massive and detailed amounts of information. The right balance between the amount of information and level of detail needed to manage RCD and the amount of effort required by survey managers to analyse it also needs to be determined. The information required for monitoring, managing and analytical purposes also needs to be better identified. Some analytical information can be seen as ‘good to know information’ that is often only required at the end of collection. After a few experiences, we were able to simplify the process by streamlining it and by identifying the proper frequency and amount of information required to monitor and manage actively a RCD. Finally, the availability and accessibility of timely paradata information that permit the evaluation of survey progress through key indicators is critical to the development and implementation of a relevant active management strategy. Without this information, it is almost impossible to build an effective active management strategy, which is at the heart of any RCD survey.

6.3 Communication

The HES and SLID experiences clearly highlighted the importance of planning and implementing an efficient communication plan. Since a RCD project is a multidisciplinary team effort that uses an extensive collaborative approach, all parties involved in collection (e.g., regional offices, subject matter, methodologist, collection manager) had to maintain ongoing communication to identify collection issues and agree on any changes to collection strategies in a timely manner. Some technical issues also impacted collection and required constant communication to resolve them and assess their impact. Finally, ongoing communication was essential to discuss how well new collection strategies performed and whether any further adjustments were required.

6.4 Staff knowledge

One of the main RCD challenges is to produce relevant, customized and manageable reports that can be easily analysed and used by the multidisciplinary team at different points in time during collection. For HES and SLID, many new key indicators and monitoring tools and reports were used to provide all necessary information to manage RCD collection. Analysing this information requires new analytical skills that will need to be developed and maintained. This experience demonstrated the need to factor in additional training needs to support and use the more elaborate active management tools that are an integral part of RCD. At the same time, the information produced for active management need to be summarized and simplified (if possible) to prevent survey collection managers and project teams from being overcome by a massive amount of information.

6.5 Technical aspects

Overall, the technical feasibility of an RCD for CATI surveys is no longer in question. As with any CATI survey, a RCD application should be thoroughly tested before the start of collection in order to resolve most of the technical anomalies. It is necessary to remain proactive and on the lookout for

other problems that may arise. Fortunately, technical problems in both surveys were very often detected as they arose during the analysis of RCD active management reports. RCD phase initiations, updates and any necessary corrective actions are generally released the day before by head office using overnight jobs. Thus, changes made were almost transparent for the regional offices except for the interviewer assignment required for each group, since the distribution of the cases among the interviewer groups can change significantly after the initiation of a new phase, especially between the initial phase and RCD Phase 1.

6.6 Interviewer staffing levels

As in many previous paradata researches, the two RCD surveys have suggested that the interviewer staffing levels are not always well aligned with the workload sample and the expected productivity. For example, the fact that in-progress units are likely to be called more often during a single day in the second half of the collection period suggests that at that point, the interviewer staffing levels are greater than the sample workload. The proportion of in-progress units attempted on the consecutive days, the rapid decrease in terms of productivity generally observed for CATI social surveys (Laflamme, 2008b, 2009), as well as the relatively large variability of the daily productivity during collection are also good indicators of some inefficiencies. In addition, the relationship observed between ‘the proportion of in-progress regular cases’ and ‘the average number of calls made for the regular in-progress cases divided by the global cap on calls’ (Figure 2) also suggested that effort was being spent on a smaller number of cases with less productivity. In that perspective, data collection managers need better interviewer staffing management and planning tools for all CATI surveys (not only for RCD surveys) to reduce some of the tension on collection productivity and costs (Couper et al., 1998) while maintaining a high level of data quality.

7. Conclusion

In the RCD context, active management represents an essential feature. However, active management for RCD faces several challenges. The first challenge for active management is to produce relevant, customized and manageable reports based on key indicators that can be easily analysed and used by survey managers at different points in time during collection. Analysing this information requires new analytical skills for survey managers that will need to be developed and maintained. A second challenge is to find the right balance between the amount and level of detail of information needed to manage RCD and the amount of effort required by survey managers to analyse it. A third challenge is to identify the information required for monitoring, managing and analytical purposes. Some analytical information can be seen as ‘good to know information’ that is often only required at the end of collection. To that extent, only the most important reports are used on daily basis to monitor and manage the RCD surveys to reduce survey management burden while the other reports are only used when required. Finally, most of our active management tools were developed with a research perspective rather than a production perspective. With more experience, it will be possible to improve the process by streamlining it and identifying the proper frequency and amount of information required to monitor and manage actively a RCD. The availability and accessibility of timely paradata information is critical to any RCD. Without this information, it is almost impossible to build an effective active management strategy for RCD.

Even though the RCD approach can be used to improve both the quality and cost-effectiveness of CATI social surveys, it should not be seen as a magic solution to all data collection issues. RCD is only one important new feature for data collection managers that need to be consolidated and used with other collection initiatives especially for those particularly interested in improving the cost-efficiency of data collection. For example, it is very important that the interviewer staffing composition (i.e., interviewer experience and skills) and levels (i.e., number of interviewers) used

throughout data collection period should be aligned with the proposed RCD strategy to take the greatest advantage of the entire data collection period.

The purpose of this paper was to describe the RCD strategy and its implementation, including the active management tools along with the highlights and lessons learned from the first two surveys used as RCD pilots.

Acknowledgements

The author would like to thank the staff in the Collection Planning and Management Division, the Regional Office Branch and Methodology of Statistics Canada, especially Milana Karaganis, H  l  ne St-Jean, Chris Mohl, Fran  ois Laflamme, Tracy Tabuchi, St  phane Dufour, Duncan Wrighte, Wisner Jocelyn and Fran  ois Gagnon for their advice, contribution and comments on this research.

References

Couper, M. P., Baker, R.P., Bethlehem, J., Clark, C.Z.F., Martin, J. Nicholls W.L. and O'Reilly, J.M. (1998), Computer Assisted Survey Information Collection, Wiley series in survey methodology section, Chapter 15 by Edwards, Suresh and Weeks, pp. 301-306.

Groves, R.M. and Heeringa, S.G. (2006), Responsive design for household surveys: Tools for actively controlling survey errors and costs. Journal of the Royal Statistical Society Series A. Volume 169, Part 3.

Hunter, L. and Carbonneau, J.-F. (2005), An Active Management Approach to Survey Collection. Proceedings from the 2005 Statistics Canada International Symposium on Methodological Issues.

Laflamme, F., Maydan, M. and Miller, A. (2008a), Using Paradata to Actively Manage Data Collection. 2008 American Statistical Association, Proceedings of the Section on Survey Research Methods.

Laflamme, F., (2008b), Data Collection Research using Paradata at Statistics Canada. Proceedings from the 2008 Statistics Canada International Symposium on Methodological Issues.

Laflamme, F., (2009), Experiences in Assessing, Monitoring and Controlling Survey Productivity and Costs at Statistics Canada. Proceedings from the 57th International Statistical Institute Conference.

Laflamme, F. and Karaganis, M. (2010), Development and Implementation of Responsive Design for CATI Surveys at Statistics Canada, presented at the European Quality Conference, Helsinki, Finland.

Laflamme, F. and St-Jean, H. (2011), Highlights and Lessons from the First Two Pilots of Responsive Collection Design for CATI Surveys, 2011 American Statistical Association, Proceedings of the Section on Survey Research Methods.

Mohl, C. and Laflamme, F. (2007), Research and Responsive Design Options for Survey Data Collection at Statistics Canada. 2007 American Statistical Association, Proceedings of the Section on Survey Research Methods.

Schouten, B., Cobben, F. and Bethlehem, J. (2009), Indicators for the representativeness of survey response, Survey Methodology, 35, pp. 101-114.

Tabuchi, T., Laflamme, F., Phillips, O., Karaganis, M. and Villeneuve, A. (2010), Responsive Design for the Survey of Labour and Income Dynamics. Proceedings from the 2010 Statistics Canada International Symposium on Methodological Issues.

Adapting Blaise and SurveyTrak for the developing world: Case study in Ghana

Elana Cohen-Khani, Statistics Netherlands

Yu-Chieh Lin, Beth-Ellen Pennell, Gina-Qian Cheung, Survey Research Center, University of Michigan

Introduction

This paper describes survey and monitoring solutions developed to conduct in-person interviews for large scale community and household panel surveys in Ghana. These innovations include a highly flexible parallel block instrument design to allow the collection of household, individual, agricultural, and business level data in an integrated fashion, as well as a case management and data transmission system that allowed researchers rapid access to a rich array of paradata. Project managers were able to make use of timestamps, keystroke data, GPS data, and respondent contact records to track interviewer performance and movement, and map local infrastructure for analyses. Adaptations were also made to address the frequent lack of phone and internet connectivity that is common in rural areas of developing countries. This paper will discuss the challenges introduced by adapting old tools to a new environment and how these solutions enabled a more effective and efficient data collection process.

Background

The EGC-ISSER Socioeconomic Panel Survey (the Panel) is a collaboration between the Economic Growth Center (EGC) at Yale University and the Institute for Statistical, Social, and Economic Research (ISSER) at the University of Ghana. The first wave of the Panel took place between 2009 and 2010 and included 5,009 randomly selected households, containing roughly 18,000 people, spread amongst 334 enumeration areas in all of Ghana's ten regions.

The current and second wave of the Panel began in late 2013 and will be finished in mid-2015. The Wave 2 sample included the original 5,009 households as well as new households formed by household members from the original sample. The Panel aims to revisit the same households periodically over the next two decades to monitor their progress across a wide range of outcomes.

Design

Survey - Blaise

The Panel contains separate questionnaires administered at the community and household levels with two separate instruments. The community level data is primarily collected before the household level data so that some of it can be preloaded into the household instrument. The community level survey has modules for schools, health facilities, financial institutions, infrastructure, land use, livestock, crop prices, crop unit conversions, extension services, shocks, employment, daily wages, district government, social and political groups, and community infrastructure projects.

The household instrument collects data at four levels: household level, individual level, plot level, and business level. The information collected at each level is:

Household	Individual	Plot	Business
Non-resident spouses	Background	Agriculture	Basic information
Non-resident relatives	Employment history	Land tenure	Assets
Durable goods	Education	Tractor and plough use	All employees
Financial assets	Migration	Seeds	Important employees
Social networking	General health	Labour	
Information seeking	Womens' health	Harvest	
Consumption	Children's module	Chemical inputs	
Housing	Psychological and social	Sales and storage	

Table 1. Household Survey sections by level of analysis

Case Management – SurveyTrak

For case management, data transmission, and data processing, we used SurveyTrak, which is a sample management system created by Survey Research Center at the University of Michigan. It has been used to manage hundreds of CATI/CAPI survey, and its strength is managing longitudinal panel surveys.

Challenges

The Questionnaire

The questionnaire takes about 12 hours to complete for households where most sections are relevant, and requires speaking to multiple respondents in the house privately. Enumerators were not permitted to spend more than two hours per day with each household. Thus, the instrument needed to be designed in such a way that sections and subsections could be accessed easily and out of sequence, and enumerators needed to have a clear view of their progress and be able to know, at a glance, what they needed to do to complete a case.

To handle this, with the assistance of staff at the SRC, we built survey layout that allowed enumerators plenty of flexibility while also restricting them to specific routing where it was necessary. We used a set of color-coded hyperlinks to give enumerators immediate feedback about their progress and also to provide quick access to each section. The main parallel tab is a dashboard from which the different sections of the survey can be accessed, and enumerators can return to this tab from any point in the survey.

In Figure 1, under the “Rosters” subheading, are links to other tabs where parallels at the person (individual), plot, and non-farm enterprise (business) level are accessible. These links only become active when the matching roster on the left hand side of the page has a “Done” status.

Survey Status			
S0Consent Forms			
S01A: Consent	Done	S01B3: Consent for Under 26	Done
Rosters			
S01B2: Household Roster	Done	Person Sections	Started
S04: Plot Roster	Done	Plot Sections	Started
S05: Non-Farm Enterprise Roster	Done	Non-Farm Enterprise Sections	Not Started
Household Level Sections			
S02B: Non-Resident Spouses	Not Started	S03Ai: Animals	Not Started
S02A: Non-Resident Relatives	Not Started	S03Aii: Tools	Not Started
S10C: Social Networking	Done	S03Aiii: Durable Goods	Not Started
S10D: Information Seeking	Not Started	S03Bi: Borrowing	Not Started
S11: Household Consumption	Not Started	S03Bii: Lending	Not Started
S12: Housing Characteristics	Not Started	S03Biii: Out-Transfers	----n/a----
S04NN: Gathering	Not Started	S03Biv: In-Transfers	----n/a----
		S03Bv: Savings	Not Started

Not all sections have been completed! Please select a section and continue the survey.

Case Complete

ivcomplete 012570003 ivcomplete

Figure 1. Main survey parallel in for Household Survey

As an example of restricted routings built into the modular system, most of the sections rely upon the household roster being complete, so most of the sections are inaccessible until the household roster is done. The Out-Transfer and In-Transfer sections rely on information collected in the Non-Resident Spouses and Non-Resident Relatives sections, so they will stay inactive until those other sections have been completed

Figure 2 shows the “Person” level of parallels, with names filled in from the household roster displayed on the left and a set of color-coded hyperlinks for each section relevant to that person visible on the right. Both of the individuals displayed below are men, so the Women’s Health sections are inaccessible.

Blaise 4.8 Data Entry - c:\blproj\ghana_p\work\householdsurvey

Forms Answer Help

HOUSEHOLD SURVEY Person Status Enterprise Status Agriculture

Name	Background	Employment	Education	Migration	Health	Womens Health	Mens Health	Children	Pysch/Social
VICTOR GBEHO	Done	Done	Done	Done	Done	---n/a---	Done	---n/a---	Done
KWASI GBEHO	Done	---n/a---	---n/a---	---n/a---	Done	---n/a---	---n/a---	Started	---n/a---

Not all sections have been completed! Please select a section to work on.

Person's Status

012570003 persontab.personlevelstatus

Figure 2. Person level tab showing color-coded hyperlinks for each person's relevant parallel blocks

Figure 3 shows the "Plot" (Agriculture) tab.

Blaise 4.8 Data Entry - c:\blproj\ghana_p\work\householdsurvey

Forms Answer Help

HOUSEHOLD SURVEY Person Status Enterprise Status Agriculture

Plot Name	Plot Info	Land Tenure	Tractors	Seeds	Labour	Harvest
FIRST PLOT	Done	Not Started	Not Started	Done	Not Started	Not Started
SECOND PLOT	Not Started	---n/a---	---n/a---	---n/a---	---n/a---	---n/a---
Chemical Inputs	Not Started					
Crop Sales and Storage	---n/a---					

Not all business sections have been completed! Please select a section to work on.

Plot's Status

012570003 agtab.plotlevelstatus

Figure 3. Agriculture (plot) level tab showing color-coded hyperlinks for each plots's relevant parallel blocks

Sample Management

Beyond the challenges of designing the survey instrument itself, SurveyTrak software had to be extended to handle situations where internet access was either poor or non-existent. SurveyTrak was already equipped to handle visit records, sending and receiving samples from the central server via the internet, and scheduling future calls and contacts.

Figure 4 shows the main screen from which enumerators can select households from the list assigned to them and start an interview. It displays the last result code, the status, the last date the record was touched, keeps a count of contact attempts, and gives the status of the contact.

RC Ind	Household ID	Result Code	IW Status	Result Date	Tel Attempts	FTF Attempts	Last Attempt Mode	6th Attempt No Contact	Cont Obs Status
	01257-0001	4202	Suspend	27/01/2014	0	1	FTF		Incomplete
	01257-0002	0000		00/00/0000	0	0			
	01257-0003	4001	Suspend	07/08/2014	0	1	FTF		Incomplete
	01257-0004	4002	Suspend	15/09/2014	0	1	FTF		Incomplete
	01257-0005	0000		00/00/0000	0	0			
	01257-0006	0000		00/00/0000	0	0			
	01257-0007	0000		00/00/0000	0	0			

Household Head: VICTOR GBEHO
 Phone: 0244-444-444 Phone Type: Cell Phone
 Read-only Phone: 0542-272-342 Read-only Phone2: 0542-272-342
 Phone2: 0277-777-777 Phone Type2: Cell Phone
 General Location: 34LE3 GBUGBLA NIIDUM FETISH PRIEST H,NII ODUM FETISH PRIEST'S RESIDENT

Region: Ashanti District: afigya kwabre Community/City: afrancho EA: House #: a/372
 Street: busia,hse opposit the fitting shop Block #: Room #: Floor #:
 Case Note:

UserID: 28811003 Name: Maxwell Atim Project: EGC-ISSER SocEco IWER Training SID: 01257-0003 Build: 12 IN 22 001 SendDate: 26/03/2015 19:47 PM

Figure 4. Sample Selection screen

At the bottom of the page is the contact information for the head of the household and an address or location. It is still uncommon in most Ghanaian communities for people to discuss locations using street addresses and numbers. Rather, people usually give directions in terms of relationships to landmarks.

SurveyTrak

Projects Backup Export Sa... Import Sam... Case Notes View/ Edit Call Wizard Planner Send/Rec Reports Find Lock Sort Conduct In... Exit

View/Edit

Call Info Respondent Address Contact 1 Address Contact 2 Address Household Level Status

Iwer	Call Date	Call Day	Call Time	Phone # Called	Mode	TEL Mode	Result Code	Contact Window	Contact Obs Status
28811003	15/09/2014	Monday	07:04 PM	- -	FTF		4002	1	Incomplete

All Notes Case Notes Save Close

UsedID: 28811003 Name: Maxwell Atim Project: EGC/ISSER SocEco Iw/ER Training ISID: 01257-0004 Build: 12.IN.22.001 SendDate: 26/03/2015 19:47 PM

Figure 5. Call Info

Selecting a line in the window seen in Figure 4 and then pressing the View/Edit button pulls up the window seen in Figures 5 and 6.

SurveyTrak

Projects Backup Export Sa... Import Sam... Case Notes View/ Edit Call Wizard Planner Send/Rec Reports Find Lock Sort Conduct In... Exit

View/Edit

Call Info Respondent Address Contact 1 Address Contact 2 Address Household Level Status

Household ID: 01257-0004

Household Head : WALLACE JONATHAN TETTEH KOFI AGGUDEY

General Location: BT/D15, ALONG THE ASUOFRI STREAM, BETOM HOUSE NUMBER BT/D15

Region: Central District: AMA Community/City: Osu EA: Osu-Machi Proci

Street Suffix: c 731/12 pig farm

House#: Block#: Floor#: Room#:

Read-only Phone: 0266-632-021 Read-only Phone2: 0246-632-021

Phone: Phone Type: Phone2: Phone Type2:

Save Close

UsedID: 28811003 Name: Maxwell Atim Project: EGC/ISSER SocEco Iw/ER Training ISID: 01257-0004 Build: 12.IN.22.001 SendDate: 26/03/2015 19:47 PM

Figure 6. Address Tab

The Call Info tab displays a list of all contact attempts for that household, the three Address tabs allow the contact information for the household head, and two contacts from outside the household, to be recorded and updated. sd

The Household Level Status tab displays the same information as the dashboard in the Household Survey instrument, which allows enumerators to see the status of each section of the survey without having to open the survey itself.

Section	Status
S01A: Consent Form:	Done
S01B3: Consent Form for Under 26:	Done
S01B2: Household Roster:	Done
S04: Agriculture:	Done
S05: Non-farm Enterprise:	Done
Personal Level:	Started
Agriculture Level:	Started
Non-Farm Enterprise Roster Level:	Not Started
S02A: Non-resident Relatives:	Not Started
S02B: Non-resident Spouses:	Not Started
S03Ai. Animals:	Not Started
S03Aii. Tools:	Not Started
S03Aiii. Durable Goods:	Not Started
S03Bi. Borrowing:	Not Started
S03Bii. Lending:	Not Started
S03Biii. Out Transfers:	Not Started
S03iv. In Transfers:	Not Started
S03v. Savings:	Not Started
S10C: Social Networking:	Done
S10D: Information Seeking:	Not Started
S11: Consumption:	Not Started
S12: Household Characteristics:	Not Started

SurveyTrak Edit Tools Help
 Projects Backup Export Sa... Import Sam... Case Notes View/Edit Call Wizard Planner Send/Rec Reports Find Lock Sort Conduct In... Exit
 View/Edit
 Call Info Respondent Address Contact 1 Address Contact 2 Address Household Level Status
 Consent
 Roster
 Section
 Save Close
 I:\edID: 28811003 IName: Maxwell, Atim IProject: FGC/JSSER SocEco/IV/ER Training ISID: 0125Z.0003 IBuild: 12.IN.22.001 ISendDate: 26/03/2015 19:47 PM

Figure 7. Household Level Status Tab

Data Transmission

Many of the enumeration areas were located in rural areas without reliable phone or internet connections, which meant that the usual Send/Receive process would not work in these situations when cases needed to be transferred between users or uploaded to the server when they were complete.

To address this critical issue, the SRC team devised a method to allow for offline transfers of cases between different computers using USB drives or SD cards. This made it possible for team leaders and field managers to collect completed cases from enumerators at their field assignments and send them back to the server when they reached an area with better connectivity. This also allowed team leaders and field managers to distribute new cases to enumerators as needed, even when they were far away from a good connection, all within the SurveyTrak system.

These accommodations made it possible for researchers to receive and review data and paradata with minimal delays, which is, arguably, one of the main benefits of computer-assisted surveying.

Conclusion

While there is a long history of Blaise and SurveyTrak, in addition to similar systems, being used for data collection in developed countries with reliable communications infrastructures, it has been less frequently used in developing world contexts, especially as part of an end-to-end solution. Using a combination of Blaise and SurveyTrak, we were able to deploy a comprehensive, efficient data collection system for a complicated survey instrument that worked effectively in an environment where that combination had not been tried before.

Technical Improvements and future directions for eCollection and multimodal data capture in the Australian Bureau of Statistics (ABS)

Author and presenter: Monica Kempster, Australian Bureau of Statistics

1. Abstract

The ABS is positioning itself strongly for increased demand for electronic collection of data, which will support the Australian Government Digital Services Strategy.¹ The ABS is further improving eCollection capability, utilization and usability for data providers. This includes adopting a self-help user model, user self-management of accounts and credentials, improvements in multi-modal data capture and management.

Significantly more surveys have been added to the electronic capture program; with many requiring new layouts to improve statistical collection, reduce modal bias, improve accessibility compliance and provide value add functions such as address validation, through the utilization of corporate geospatial services. Currently the ABS has been migrating forms to the eCollection environment under a 'Translation Phase' which means that the layout and questions are created as a direct copy of what is currently sent out under our paper or personal interviewer modes. As the ABS moves into a transformation phase for its acquisition of data, the ABS will take opportunities to utilize the new eCollection mode to enhance data collection through the use of web Services, layout questions to be more usable in a browser or device format, and potentially provide users with the opportunity to provide more data than what was previously collected due to paper form constraints. The use of Blaise in the transformation space for data collection, editing and analysis is instrumental to the success of this phase of the project. Through the work done over the last 12 months, an increased focus on multi-modal data capture and processing has become a core part of development and improvement.

The ABS is also adopting a new corporate authentication and authorisation solution. Changes have been made to the existing Blaise solution to improve connectivity with other components, such as authentication, security and processing services. There are also improvements in system security, load and performance. This work will make the Blaise instrument development and linkage more robust for the future, through the exclusion of all non-form processing services to servers outside of the Blaise server park environment and making the instrument more resilient to external ICT changes.

Future improvements include the ability to move partially completed records between data collection modes, making multi modal capture more timely and adaptable to changing data provider behaviors, as well as enabling scale-out of infrastructure to multiple Blaise parks for load balancing, risk management and contingency planning.

2. Background

The ABS has made a strategic decision to use Blaise IS for all web form development, as well as continuing its use for the interviewer collected household collections. Current web form development is using Blaise 4.8.4 (upgrading to latest builds approximately every 9 – 12 months). The ABS released the first electronic form (eForm) into production in December 2012, and has continued a migration work schedule since then, moving both business and household forms into the environment.

Across the last two years a number of requirements have come up for adding additional functionality as more complex forms are migrated. These have included requests for more complex layouts (to support matrix style responses) and enrichment of feature sets through the use of external services. External services such as the ABS Address Validation service, which has been linked into the form, to provide address validation in Agriculture forms. This will improve the ABS ability to utilize geospatially enabled data in publications and micro level datasets.

The ABS has also been developing a corporate Authentication and Authorisation solution to be utilized across all externally facing systems. The External Identity Access Management (XIAM) solution will be first utilized in the eCollection space. It will move users to a system of self-managed credentials, as well as providing features to update their contact information. The XIAM system will also provide obligation management functions such as delegation of forms between registered users and a dashboard to show total obligations and their statuses. XIAM will also deliver the ability to group a number of form obligations into a single set for management, which will improve the management of data providers with large numbers of forms and significantly improve the migration of surveys that use sets of forms to collect information.

The XIAM solution will decouple Blaise from authentication and authorization services. This will provide the ability to utilize multiple Blaise parks to produce a technical solution which enables the ABS to:

- scale out performance
- reduce complexity
- manage surveys
- exclude critical or high volume surveys onto a separate Blaise park, and
- manage external components such as provider portals in a more modular way.

External factors such as compliance with Accessibility Guidelines⁴ which is expected of all Australian Government agencies are also generating changes in the delivery of eForms. The ABS has made some improvements to both our code and the 'out of the box' CSS, however, there is still more to do be done to meet the expected level of compliance.

3. Current Architecture

The following diagram shows the current architectural solution overview for the Blaise eCollect platform. The diagram depicts all major solution components as well as major communication flows.

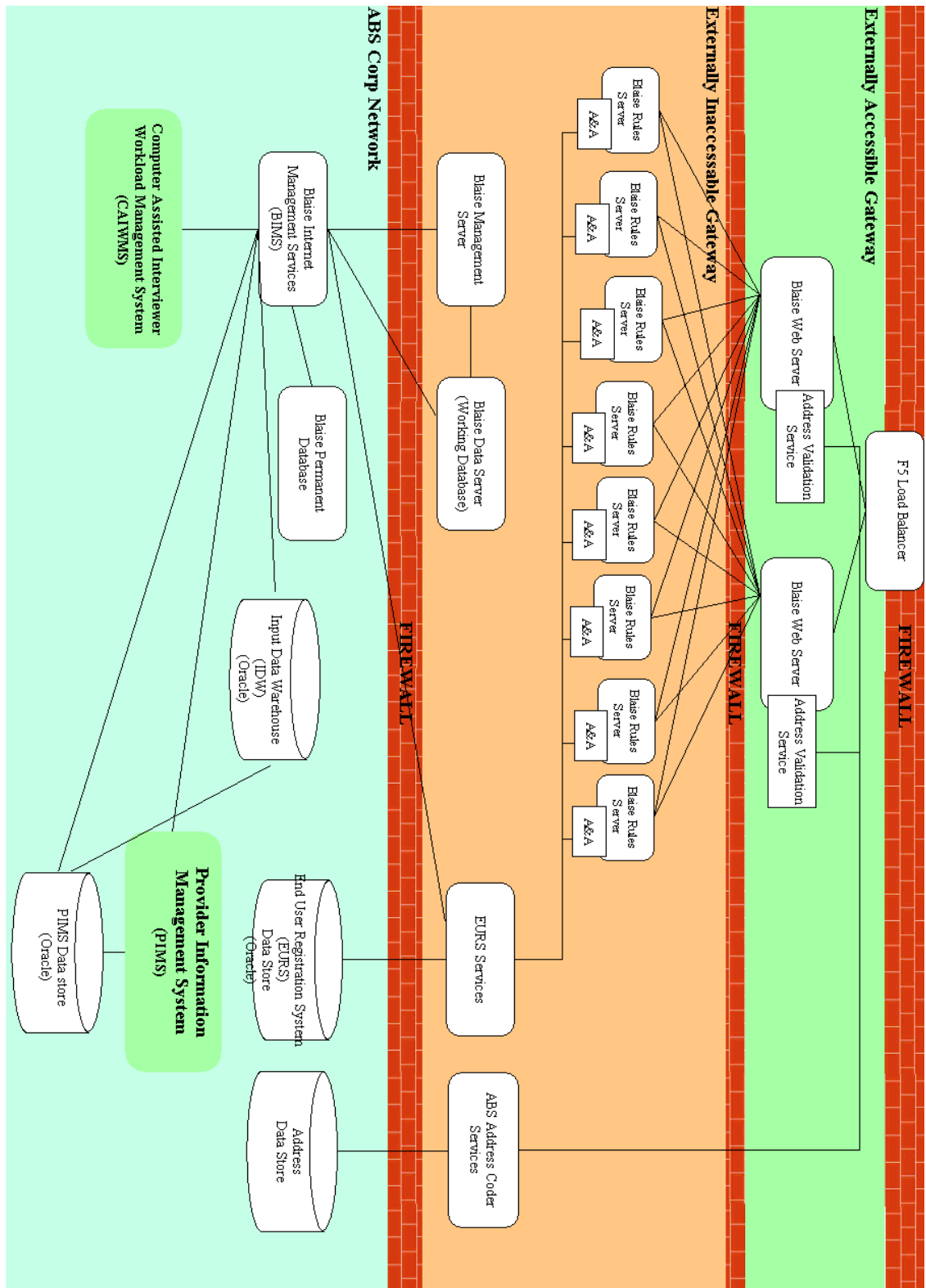


Figure 1 Current Blaise eCollection Solution Architecture

3.1 Components

The ABS ECollection solution utilizes the ability to put the Blaise IS components onto multiple servers to improve performance by utilizing multiple server capacity. The number of scaled out servers for each component has been determined based on load testing, and optimized to maximize the throughput and stability of the system.

The current environment configuration of 2 Web Servers, 8 Rules Servers and one Data server is what ABS found to be its most optimal configuration through load testing, for more details see the 2013 IBUC Paper, Blaise 4.8 Load and Performance Testing². This architecture maximizes the total number of connections which can be handled by the Data Server (of which there can be only one in a park) while ensuring that no other component experiences performance degradation. As can be seen by the fact that there are 8 Rules servers, this server experiences the highest load of all Blaise servers.

3.1.1 Blaise Web Servers

The Blaise Web Servers provide the presentation layer for Blaise Internet. The Web Servers interact with the Rules servers when a user navigates to the next survey page, when a critical field is updated, or a user submits a survey. We also have the Address Validation Service installed on each of the Webservers.

3.1.2 Blaise Rule Servers

The Blaise Rules Servers are responsible for executing the business logic for web surveys. For example, determination of the next sequence of questions and compute derivations. It also invokes the authentication and authorization DLL calls via alien procedure calls within the instrument.

3.1.3 Blaise Management Server

The Blaise Management server coordinates the Blaise Server Park for load balancing, and provides functionality for deployment and management of surveys deployed to Blaise Internet. In the ABS production environment all Blaise Management Server functions are accessed via the BIMS Services (discussed in more detail in section 3.1.6). The current architecture also means that the Blaise Manager is within the gateway environment, and not accessible to anyone other than a few members of the ABS Technical Infrastructure team. By utilizing the BIMS Services (which are in the corporate environment) other users of the Blaise eCollection systems can invoke Blaise Management Server functions from within the corporate environment. This has been done in line with our security policies and architecture principles.

3.1.4 Blaise Data Server (Live Database)

The Blaise Data Server (Live Database) stores collected provider data for the duration of a survey; this includes both partially completed responses, as well as submitted survey data. Completed survey data is then copied (via BIMS) into the Permanent Data store. Access to this database is only via the BIMS Services for the same reasons described in 3.1.3 above.

3.1.5 Blaise Permanent Database (Offline Database)

The Permanent or Offline data store is used to store provider response data once it has been submitted. This component is used to mitigate security risks associated with storing provider data permanently in the live data store (see 3.1.4), due to the fact that it is in the gateway environment. It also reduces load risk; as more intensive data management functions such as load to our Input Data Warehouse (IDW) do not have any impact on the performance of the online systems. This database is also within the corporate environment which makes it more accessible to other systems, and functions.

3.1.6 Blaise Internet Management Services (BIMS)

The Blaise Internet Management Services (BIMS) is an ABS built web service layer that provides interfaces to survey lifecycle management functions in the Blaise API, and links to PIMS, CAIWMS and Authorisation functions.

The Blaise Internet Management Services provide a single point of access for other system to access Blaise Functions. This has been done to provide a ‘wrapper service around Blaise’ to enable the Blaise API calls to be obscured from other systems, and provide a consistent interface access for other systems, and limit changes which may be required to these services when Blaise upgrades occur to a single place within the environment.

Operations that are facilitated by BIMS are deployment include

- deployment, update and decommission of surveys
- update of prefill data
- retrieval of processing statuses, and
- access to provider response data and Blaise journal functions.
-

BIMS interacts with other ABS systems such as

- the Input Data Warehouse (IDW),
- Provider Information Management Systems (PIMS),
- External User Registration System (EURS) and
- Computer Assisted Interviewer Workload Management System (CAIWMS).

The PIMS and CAIWMS systems provider the user interfaces for people to invoke the BIMS Services such as initialize survey, prefill data retrieve data and access journal data.

3.1.7 External User Registration System (EURS)

The External User Registration System (EURS) is an ABS bespoke developed system, which uses an Oracle database and JBOSS developed services to provide Authentication and Authorisation functions. This system is used for the ABS eCollections and dissemination products. EURS is currently utilized in the eCollection environment to provide a pre created user identifier and password to eCollection data providers, and already linked survey obligations.

3.1.8 Authentication and Authorisation Module

The Authentication and Authorisation Module is an ABS developed component. Its purpose is to provide the link between Blaise and the EURS Services, given that Alien Procedures will only call .Net, it effectively provides some business logic and a wrapper for the Java based services. It is installed on each instance of the Blaise Rule Servers. It is used for Authorisation services, as well as to write back the completion status of a form to the EURS data store

3.1.9 Provider Information Management System (PIMS)

The Provider Information Management System (PIMS) is used to manage business surveys within the ABS. It contains the selected units, response statuses, workflow management for Intensive Follow Up (IFU) and sample management. It is also used to initiate and document provider contact information for the management of surveys. It is used in conjunction with the Input Data Warehouse (IDW) which stores the provider responses. There are other internal systems which are used to manage this further, but will not be discussed in this paper.

3.1.10 Computer Assisted Interviewer Workload Management System (CAIWMS)

The Computer Assisted Interviewer Workload Management System (CAIWMS) is the ABS system to manage Household Surveys. It contains the interviewer workload allocations, response statuses, links to coding systems and the respondent data, which is cleaned, coded and then extracted. This is where the survey initialisation for household forms is done, through the CAIWMS interface, linking with

BIMS to create the user registrations in EURS, and copy prefill data into the online data stores for reference when someone accesses their forms.

3.1.11 Input Data Warehouse (IDW)

The Input Data Warehouse is an ABS developed solution, built on Oracle to hold micro data for all business surveys. This data store is loaded with the eCollection information for business surveys via a BIMS call to the IDW loader.

3.2 Current Challenges and Limitations

Discussed below are some of our key challenges in the architecture and system environment, which we are currently addressing.

3.2.1 Authentication and Authorisation

Briefly, Authentication is providing a way for someone to prove who they say they are, usually through the provision of a user identifier and password matching combination. Authorisation is what an authenticated user has access to within the environment.

The eCollect authentication and authorization is currently a bespoke system, which uses EURS to provide Authentication services, and an Authorisation system developed in Blaise to persistently store form obligations, and to provide the user interface for both Authentication and Authorisation functions.

There were a lot of additional checks put in place within Blaise to ensure that the security of the system was maintained, and things like session and URL tampering could not result in a security breach. While this solution has served for the last two years in enabling the ABS to initially move to eForms, the desire for a more featured user portal and whole of organization approach to External Registration and Authorization has led to the development of a new system.

Reasons for this are discussed briefly below.

- The current method of providing businesses or households with a user identifier and pre generated password means that the ABS does not necessarily have a link to the person filling out the form, only the address of a household, or the primary contact at a business (who may not be the data respondent)
- Providing a user with an identifier and password means that if the password is forgotten, or the account is locked, there is no meaningful way to provide a challenge/response check (i.e. secret question and answer, or email token) to provide a secure way of identifying the authorized requester to reset a password
- Use of already existing components in industry is now the preferred approach to the creation of new ICT systems within the ABS, reducing our dependency on bespoke systems, such as our internally developed Blaise solution
- For large ‘fleets’ of responses, such as Survey of Motor Vehicle use, a data respondent could have 30-40 motor vehicles to respond for. At present this has not been moved onto eCollection, as the current method of providing 30 ids with 30 passwords (one for each obligation) is not seen as an appropriate end user experience
- The ability to provide a ‘portal’ experience to users, where they can see all of their obligations, their due dates and status, is expected to improve users ability to manage their responses in a timely, consistent manner
- The use of a separate authentication and authorization solution will improve security of the system, as newer technologies can be used to ensure that URL tampering and session tampering cannot be used

- Moving authentication and authorization services away from Blaise will reduce the load on the Blaise servers, improving load performance of the Blaise services.
- People can create a persistent account to re-use credentials for additional survey obligations – including a way to link both business and household obligations into a single
- The ability to delegate an obligation to another registered user, currently the only way that can be done is by sharing credentials, which creates a security risk, and obscures management information which would assist the ABS in understating user behaviors when completing survey obligations
- Use of a single solution across the ABS for Authentication and Authorisation means when security requirements change (e.g. password lengths, or complexity) a single system can be changed)
- The solution should be more scalable than the previous solution
- The removal of authorisation from the Blaise environment opens up opportunities to have multiple Blaise parks, as the authorisation information will be in a data store which isn't within the park.

3.2.2 Data Retrieval of only Completed forms

Currently systems are only designed to retrieve submitted data, as our legacy systems were not updated to handle the retrieval of Web forms which are partially complete, and moving them into an interviewers' workload, this opportunity has come about through the utilization of multi modal data capture. While services could be built to retrieve any data, challenges in legacy systems have made this a future enhancement, rather than a current functionality.

The business challenge to manage while this technical limitation exists is that respondents, who may have significantly completed a form online, will be re-asked all of that content again by an interviewer, possibly increasing respondent frustration, and not maximizing the use of interviewers to only complete the data which hasn't been already collected.

3.2.3 'Plug and Play' Address Validation

The current implementation of address validation was an initial technical delivery. At present it is tightly coupled to the name and position of the Blaise field within the instrument, which means that each instrument which wants to implement this service either has to put it in the exact same named field, with the same designed page, or we need to add those field names into the Validation Services. Work is being done to decouple the exact Blaise name from the call and create a plug and play module, so that it can be packaged into the Blaise Instrument and use a generic service, which can be referenced on a separate server, rather than depending on it being deployed on the Blaise Web Servers to enable communication

4. Proposed New Architecture

The architecture discussed below is currently being developed, with most of it in user acceptance testing. It is expected that this will be the production system within the next 6 months

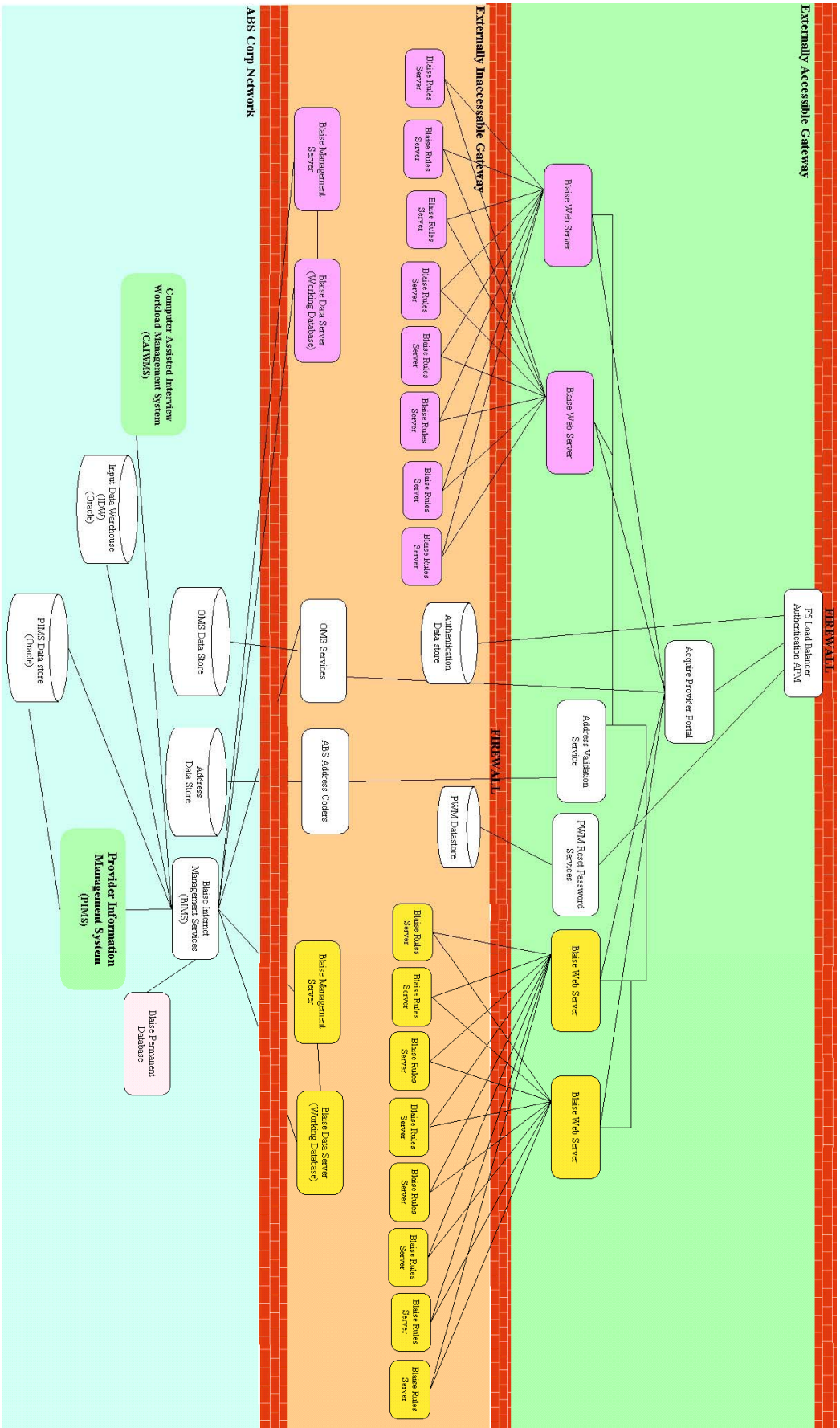


Figure 2 – Future Architecture of eCollect

Notes:

Mark Complete will be a service called from the ASP Blaise Pages, rather than from the Instrument

Links of the Blaise Management server to all other Blaise Servers (Web, Rules) have been removed to improve readability of diagram

Purple Park is a single Blaise Park. Yellow is another. Orchestration of dual parks is done through the single instances of the OMS Services, and BIMS Services, each of which would contain metadata to determine which park installation is on

4.1 Components

The discussion below focuses only on the improvements made to the system. Given that it is expected in the short to medium term for the ABS eCollection to run on Blaise 4.8.4 the dimensionality of the Blaise server park has not changed. The details of some components have been removed from this section as there have not been any changes made.

4.1.1 Blaise Web Servers

The Blaise Web Servers will be set up the same as the current architecture; however, the Address Validation Services will be moved onto a separate server. This will move the load which is currently being managed by the Blaise Web Servers onto a separate server, which means that non eForm use of those services won't have any impact on the Blaise Server performance. It will also make deployment of newer versions of the Address Validation Services easier, as it won't require taking down part of the eCollection environment to update the service.

4.1.2 Blaise Rule Servers

The Blaise Rules Servers will have the Authentication and Authorisation DLL removed, and it will be replaced with a Web Services which will be called from ASP.

In the short term for Initial production release of XIAM, an OMSServicesUser.DLL will be put in place this will write survey completion statuses to the OMS store. It will be deployed in the same manner that the current Authentication and Authorisation DLL is architected.

4.1.3 Address Validation Services

The address validation services are effectively gateway enabled wrappers to enable linkage with internal ABS Services, to enable them to be consumed by Blaise within the gateway. These have a potential future user base much larger than just eCollection, including geospatially enabled data, Census data searches via address, and National Regional Profiles. These address services will be rewritten to de-couple the Blaise names and field indexes, to enable all Blaise Instruments to use a plug and play approach to calling these services. The movement of these services onto a separate server in the gateway will also make them more usable broadly in the ABS.

4.1.4 Obligation Management Services (OMS)

The Obligation Management Services (OMS) is replacing some EURS components of the old system which will store the obligations that a user has. Storage of Authentication and Authorisation data has been moved into separate data stores for the new release, allowing authentication to be managed as a Corporate System, and authorization (Obligation Management) to be built for each application depending on unique requirements.

OMS uses an oracle database which is configured in the corporate domain, and provides the ability for the acquire portal to make changes to obligation records in the data store via this service.

4.1.5 PWM

PWM is an open source password self-service application³; it uses an LDAP data store and provides a basic set of customizable password management features. This has been chosen as a 'buy before build' option to move the ABS into our future method of building systems. It provides the setting of secret question and answers for credential management, reset of password, and forgotten password services.

4.1.6 Access Policy Manager (APM)

Access Policy Manager or APM is a F5 device plug in which provides the ability to manage authentication (i.e. credentials to allow access) this provides the create account and login functions. This solution promises to be highly scalable, available and be able to be reused across the ABS.

4.1.7 Acquire Provider Portal

The acquire provider portal has been built to provide the interface layer where users can link an obligation, access a dashboard of their obligations, update their email address and call change password functions. They can also delegate obligations to other authorized users.

From the obligation dashboard a user can then click on the survey link and access the Blaise Instrument in order to complete their response and submit their form.

4.1.8 Multiple Server Parks

The removal of the Authorisation being in a Blaise Instrument will mean that with some changes to BIMS deployment metadata (to enable it to capture which park an instrument is deployed in) the eCollection system will be capable of managing multiple Blaise parks. This will be useful as a means to be able to handle additional load, by increasing the capacity of the eCollection Blaise Instrument component, as well as provide the ability to be able to exclude particular surveys onto a separate park (for example the Agricultural Census scheduled to be delivered in 2016). At this point there is nothing which suggests that extension beyond 2 parks is not possible, however load limits on the BIMS services would need to be tested and consideration given to a load balanced pair of BIMS Services.

5. Brief overview of Blaise Instrument architecture

The way instruments are architected allows for information which is already known about the form, such as the selection and historical information. This information is known as prefill data to be utilized within the form. For a diagram depicting the relationship between Blaise source code and compiled packages see Figure 3

The BIMS services are used to load the prefill data from either the PIMS environment (Business Surveys) or CAIWMS (Household Surveys) and it is written into the Working database server, as a _INIT named database (for each survey).

For some surveys data is loaded from CSV files into databases which are packaged into the instrument. This is done for surveys where we have information from other bespoke systems, which is required to provide data respondents context of what they are to report on, for example building jobs for the Building and Construction Survey.

The instrument package is deployed via BIMS, the internet BIP file is used to deploy the instrument out into the Blaise server park, and the other version of the code is used to create the metadata for CAIWMS. BOI files are also created to enable BIMS to access the INIT, Journal, Permanent, Temporary and Work data stores

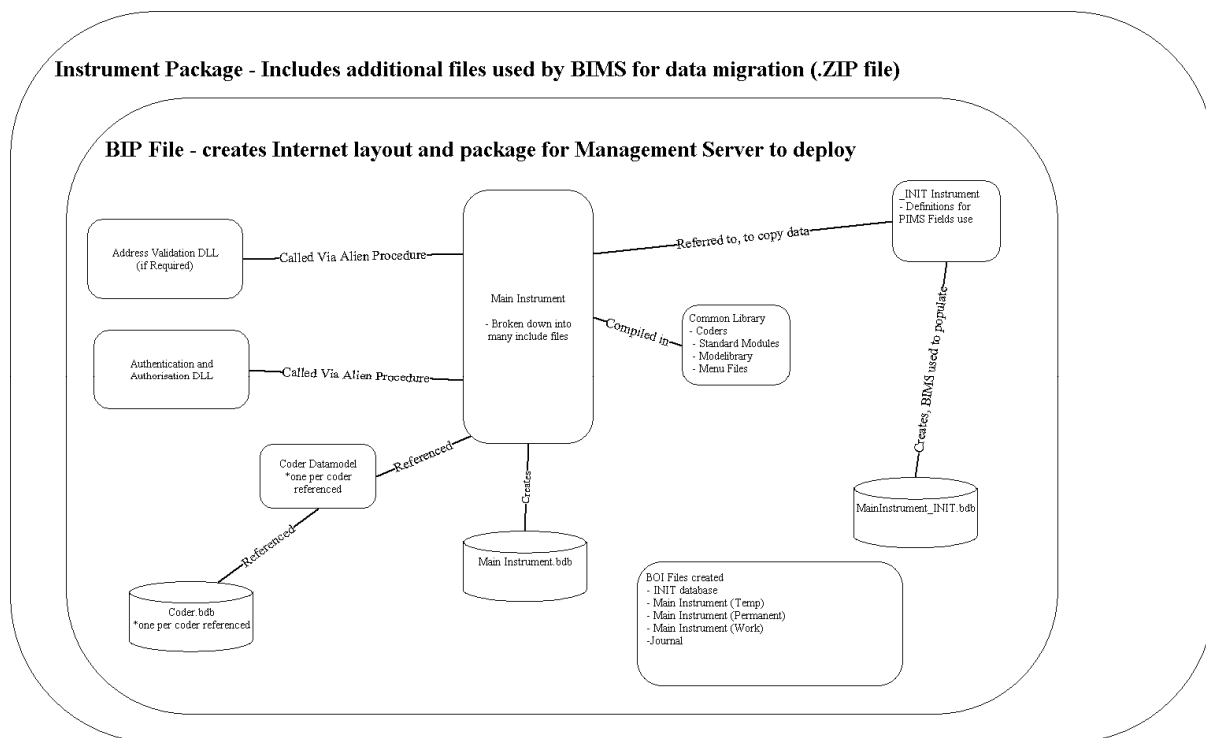


Figure 3 – Blaise Instrument Architecture

6. Accessibility Compliance - Issues, Challenges and Improvements made

There are still a number of challenges which are not directly to do with the large scale architecture. As discussed before, all Government online systems are expected to conform to WCAG 2.0 AA Standards. While the ABS has made some changes to bring us closer to conforming with these standards there is still more work to be completed.

There has been a tension between usability and accessibility. Some of the decisions made to improve the layout and usability of the form have had a negative impact on the ability of the form to be read and understood by users of screen readers. For example improved layout has decoupled the question wording from the data entry field, which provides a better looking interface, but makes it more challenging for screen reader users to match the question with the response field.

6.1 Accessibility Compliance Issues and Challenges

Some issues which have been raised through Accessibility Testing which we have currently not solved are:

- Menus, Headers and Footers – they are not marked up as lists, so are not able to be easily navigated
- Resize of text using Browser zoom
- Pages sharing the same title – each page uses the Data model name for the title
- Change of context – occurs when rules are re-run on a page (currently happens on re-derive of field, or on error being triggered)
- Error message not being read by screen readers
- Label_ids are not meaningful but are used by screen readers
- HTML code does not validate appropriately (when checked with W3C Validator)
- Repeated content in headers and footers

- Radio button selection and check box selection is not consistent in whether you can click on the line to select

6.2 Accessibility Compliance Improvements Made

Changes which have been made which have improved accessibility compliance

- The user of the layout=presentation for tables which are implemented for layout purposes only
- Text, rather than text as images on buttons
- Alt text has been added to all images
- Improvement of layout of forms so that screen readers keep context together
- Using Header Markups to enable users to scan through the page to provide a structure which can be quickly navigated
- Colour Contrast
- Role of pages is no longer set to 'application'

In order to continue to improve our compliance, further training of staff to understand the guidelines, utilize tools for assessing compliance and understanding of how to use tools like JAWS for screen reading will assist.

7. Management Information

An e-Form flag has been added to the Labor Force output data, to enable analysis to be done based on the mode that the data has been collected in. This is a first step toward having more management information, processing statuses and multi modal management available.

This flag has been enabled due to increased divergence between CAWI and CAPI Instruments, in respects to question wording, questions asked of respondents, sequencing of questions, derivations and fields which are on path.

The reason for having an e-form flag available in the environment is to:

- provide the ability to analyse the data, and determine if there is any modal bias in responses to the instrument
- Manage which edits within the instrument are triggered for eforms when they are bought into the processing environment (i.e. don't run CAPI only rules which aren't triggered in the CAWI Instrument)
- Provide abilities to select different data items, or processes based on the mode of collection

We are currently only in early stages of implementing this, given that the CAIWMS is a legacy system, and changes can be difficult to manage effectively. It is also important to ensure that changes are not going to adversely impact the processing of data.

8. Alerting, Monitoring and server restarts

There have been instances where the ABS has experienced instability in its systems. Due to this there are now regular restarts of all servers, to ensure any memory leaks, or threading issues are regularly reset in the environment.

Some of these restarts have been put in place as a temporary measure to enable the ABS to thoroughly investigate the issues, and be assured that the fix is operational before making any changes to the production environment.

Automated test scenarios and alerts have also been set up on the servers, and a number of ABS staff across Technical Applications, Technical Infrastructure, Household Instruments and Business

Instruments are now alerted should an outage occur so a rapid response can be taken, and impacts and actions can be managed.

9. Modular Design

The desire of having ‘plug and play’ components within the ABS is also extended to Blaise instruments. The ability to quickly deliver instruments to collect various different topics, and assemble already created components, rather than delivering an instrument from a new specification will assist with driving the ABS forward with the goal of fast turnaround on statistics improving time to field, use of already tested questions and sequencing and known working components.

This is driving forward the requirement to build instruments in blocks and assemble, rather than having large, monolithic Blaise instruments, which are built for a single purpose.

Having components, such as the Address Validation Tool, as plug and play modules will also further this capability, and make linkages to rich services into instruments a more straight forward approach, and the technical complexity obscured within the black box component.

As part of the Acquire transformation phase the ability to manage obligations which have a dependency order for completion and provide seamless transition between these components is being investigated. An example of this would be the Monthly Population Survey which for a respondent in a given month may contain both the Labor Force survey, as well as a supplementary survey, both of which are required to be answered, with the Labor Force needing to be completed before moving onto the supplementary, as data is transferred and referenced in the supplementary from the Labor Force.

10. Summary

Significant work has been done to fine tune the eCollection infrastructure to improve scalability, performance and reuse. The current work to improve the Authentication and Authorisation solutions will provide a number of benefits such as improved user experience, encourage self-management of accounts, and obligation management, and also provide significant opportunities to further scale out the eCollection architecture to continue the migration of ABS Instruments into online data capture.

The ABS will continue to work towards improving Accessibility compliance, both through Blaise Coding improvements, CSS templates and looking toward future Blaise releases to assist in providing opportunities to make changes to improve conformance.

Future work to improve modularity of Instruments is progressing to allow the ABS to adapt and meet new demands in a rapidly changing environment.

11. References

1. Australian Government Digital Strategy, Accessed 16/03/2015, <https://agds.test.govspace.gov.au/dss/>
2. Oleg Volguine, Blaise 4.8 Load and Performance Testing, 2013, Accessed 22/03/2015, <http://www.blaiseusers.org/2013/papers/2013Proceedings.pdf>
3. PWM, Accessed 22/03/2015, <http://code.google.com/p/pwm/>
4. WCAG 2, Accessed 22/03/2015, <http://www.w3.org/TR/WCAG20/>

A CAPI Logistic System

Gerrit de Bolster, Statistics Netherlands

1. Introduction

As of October 2010 The Netherlands consists of European Netherlands and three islands in the Caribbean: Bonaire, St Eustatius and Saba, also called The Caribbean Netherlands (CN). Statistics Netherlands (SN) is therefore also responsible for statistics in CN and for this reason runs a small office on Bonaire, the largest island of CN. From 2013 this office started using Windows tablets with a Blaise tool for data collection. In 2013 it was still limited to a tool with its own shell on each of the tablets. As it was expected to expand in the future a new system was developed. This system makes it possible for the involved departments in The (European) Netherlands to easily maintain the instruments and survey cases on the tablets in connection with the group of interviewers involved. The first questionnaire to use this system was the Labour Force Survey (LFS) which was specially designed to work on these tablets. The LFS was deployed in the field using the new system in the beginning of October 2014. This paper will give more insight about that system, the LFS questionnaire and experiences in the field in CN.

2. History

2.1 A Proof of Concept

In 2011 as a spin-off of the Case Management working group which is focusing on the requirements for a CAPI case management system, an initiative of the BCLUB, at SN a begin was made with the creation of a Proof of Concept (PoC) of a so called CAPI Case Management System (CCMS). The PoC included an application that was aimed at running on a laptop of the interviewer. Additionally a very simple application was created to run in the office to communicate with the application on the laptop. This PoC was presented during the IBUC 2012 in London.

2.2 The Omnibus Survey CN

In the beginning of 2013 the office of SN in The Caribbean Netherlands (CN) decided to use tablets to replace their paper based data collection. For that reason they requested the creation of a questionnaire for the Omnibus Survey to be held only on Bonaire, the largest island of CN. As it was an opportunity to get experience with Blaise on tablets the request was granted but considered as a PoC. It was also necessary to create a shell program to select the survey case at the opening of the questionnaire and to export the completed statements to a USB-key as the tablets were not connected to the Internet for security reasons. This shell program was built in Maniplus in combination with VBScript and some small freeware programs. This software (shell + questionnaire) was installed on the tablets using a MSI package. The survey cases were also included in the installation package packed in an encrypted zip-file. To send the exported completed statements to the SN office in The Netherlands through secure data communication from a laptop existing upload functionality through HTTPS was used. The survey should run on Dell tablets with Windows 8. The screen size was 10 inches with a resolution of 1366x768. Therefore the questionnaire was designed with a font size of 18 and a special menu including a numeric keyboard on the right hand side. The Omnibus Survey was held in June 2013. The tablets were working well to the great satisfaction of the interviewers and the local office. The results were presented during the IBUC 2013 in Washington DC.

2.3 The Consumer Price Index

Because of the success with the Omnibus Survey our office in CN wanted to continue with CAPI on tablets. Their next target was the monthly Consumer Price Index (CPI). Until then the interviewers (4 on Bonaire working 2 at the same time and 1 on Saba and 1 on Sint-Eustatius) went around the shops with a paper form printed locally on the islands that they copied afterwards to an Excel spreadsheet at home. This spreadsheet was then sent to the office in The Netherlands by e-mail. It was a troublesome and error prone process. So again the office in CN asked for a solution for the tablets. In this case a Maniplus setup for the CPI was built as the demanded functionality was not suitable for a standard Blaise questionnaire. Again an additional shell program was built similar to the one for the Omnibus Survey. Only this time it should not only export completed statements but also import files with new products for the next month. These files were transported from our office to CN using the existing functionality of download portals. The results were uploaded using the existing HTTPS functionality as before. To install the software on the tablets a new and more flexible installer (also built in Blaise) was used. The CPI on tablets started being used on Bonaire in April 2013, in August 2013 on Sint-Eustatius and in November 2013 on Saba.

2.4 The Economic Cycle Survey

At same time as the CPI instrument was being developed the office on Bonaire started to develop a Blaise questionnaire for the Economic Cycle Survey (a survey amongst enterprises asking them about their business expectations for the near future). Having no training in the use of Blaise whatsoever they soon called for help. Finally the questionnaire was mainly developed in SN. Again a separate shell program was added and the Blaise built installer used. Also the same data communication solution (HTTPS upload) was used again. Running in November 2013 it was again successful.

2.5 A new system for CAPI

Although the data collection process applying tablets was successful the office in CN was missing functionality to follow the progress. Also the repeated installation of the software on the tablets for each survey started to be a burden and the data communication process required several manual operations. Moreover there was not enough capacity available to continue the building of the individual shell programs. As it was clear that CN wanted to continue to use the tablets for data collection at the end of 2013 an informal project was started to design and build a complete new system for CN in Blaise 4.8.4. The new system was put in action on Bonaire in October 2013 and on Saba and Sint-Eustatius in November 2013 with the questionnaire for the Labour Force Survey specially built to operate with this system and tablets. As it was a first version it was considered as a pilot.

3. The architecture

3.1 Overview

The system consists of 4 separate applications mainly built in Blaise 4.8.4 including some additional VBScript (encrypted: .vbe), HTML application (.hta) and freeware programs. It supports the deployment of instruments, survey cases and additional data to interviewers (optional via local agents) and collecting the completed or cancelled statements. It also contains a messaging system for human readable and

system messages. The configuration of the software involved can also be managed remotely. Data communication is invoked using separate scripts and can be adjusted to the local needs. Any data that is exchanged is encrypted and thus protected.

Although this kind of system is generally called a CAPI Case Management System it is not really managing cases. Its functionality is clearly logistic functionality. Therefore the name “**CAPI Logistic System**” (CPLS) fits better.

3.2 The roles

In the scope of the CPLS we recognize 4 roles:

- **The main office**
The main office can manage directly one or more local agents (or none), one or more local coordinators (or none) and one or more interviewers (or none). The main office should at least manage a local agent and/or an interviewer.
- **Local agents**
A local agent can manage one or more local coordinators (or none) and a group of interviewers.
- **Local coordinators**
Local coordinators only play a role as hub between 2 transport modes. E.g. between an Internet connection with the main office and USB-keys used by a group of interviewers.
- **Interviewers**
The interviewers collect the data by interviewing the respondents.

3.3 The applications

The CPLS consists of the following 4 applications:

- **BLS-Collector**
This application runs on the network (or computer) in the main office of the data collector. The complete CPLS is controlled (directly or indirectly) by it. Instruments and Survey cases are imported in BLS-Collector before they are sent to the interviewers. The completed statements are received by BLS-Collector before they are distributed to the survey departments.
- **BLS-Agent**
The BLS-Agent application runs on the computer of a local agent. A local agent is an in-between between the office and a group of interviewers. It receives instruments and survey cases from the main office sent with the BLS-Collector application and divides them over the interviewers it manages. The completed statements received from these interviewers are sent directly to the main office. This action is registered in BLS-Agent so the local administrator always has an overview of the progress of the data collection by the group of interviewers it manages.
- **BLS-Connect**
BLS-Connect serves as a connection hub between the main office on one side and one or more interviewers on the other side or between a local agent on one side and a group of interviewers on the other side. With BLS-connect 2 different types of data communication can be connected. The main office can e.g. use a Secure FTP (sFTP) connection to a local coordinator who can copy the data to USB-keys used by operating interviewers in the area and vice versa.
- **BLS-CAPI**
With the BLS-CAPI application running on its laptop or tablet an interviewer can manage and perform the assigned interviews and communicate with the local agent or the main office using the messaging function. A laptop or tablet can be used by several interviewers each with his or her own working space.

Figure1. The main dialog of BLS-CAPI.

Questionnaires

Code	Name	Duration
AKO_CN	Labour force survey Caribbean Netherlands	1:30
BEO_CN	Visitors exit survey	0:15
LeisureSurvey	Leisure survey	0:10

Addresses (AKO_CN)

Sortorder: ZIP-code ☒ Show addresses sent

ZIP-code	Address	Name (telephone)	From	Untill	Appointment	Status
3005-110	BELLEVUEROAD 2F	Medill, Mary				[opened]
3005-110	BELLEVUEROAD 4F	Taylor, Joseph Frank				
3005-110	BELLEVUEROAD 7F					
3005-110	BIG PASTUREROAD 1F	Wesley, Patricia Winifred				
3005-110	BIG PASTUREROAD 3F	Tolboom, Johanna Grada				
3005-110	BIGSTONEROAD 0	Palermo, Joey Charles				
3005-110	BIGSTONEROAD 5	Pharo, Craig Forbes Tempest				
3005-110	FRENCH LEAVEROAD 0	Zagers, Dean Stuart				

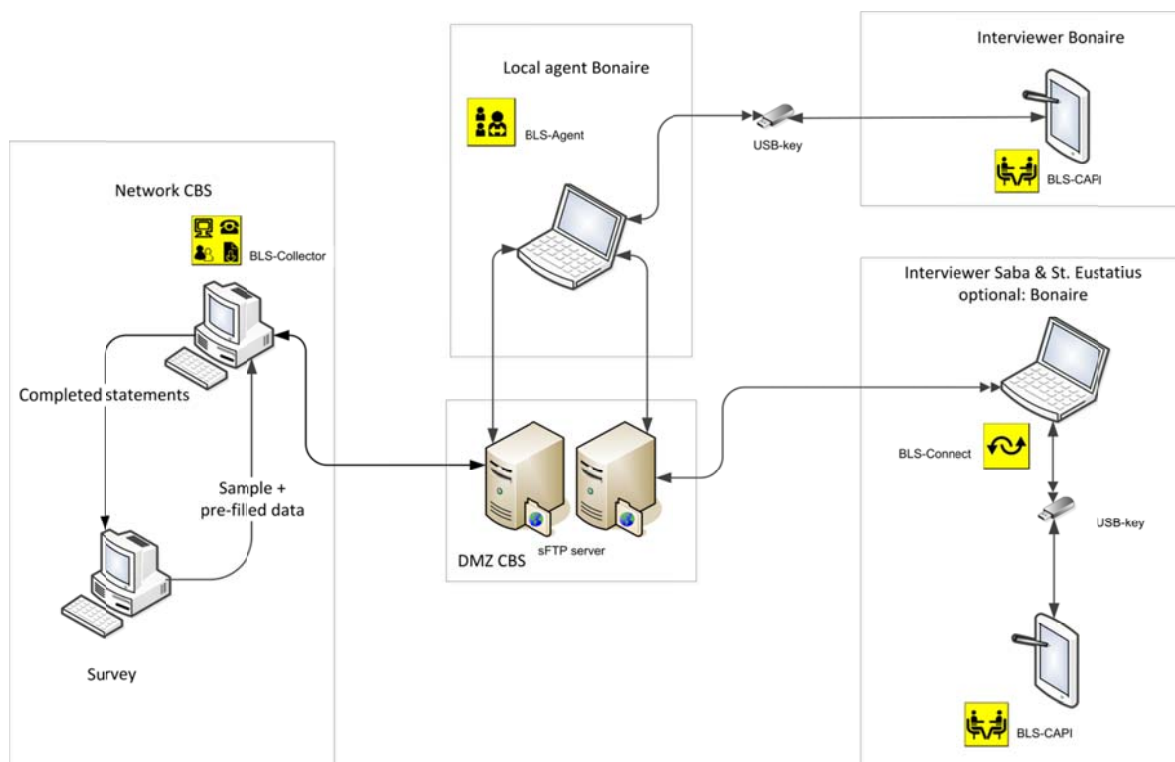
The 4 applications can be combined in different configurations. Any combination of these 4 configurations can be applied in the CPLS.

1. BLS-Collector — BLS-CAPI
2. BLS-Collector — BLS-Connect — BLS-CAPI
3. BLS-Collector — BLS-Agent — BLS-CAPI
4. BLS-Collector — BLS-Agent — BLS-Connect — BLS-CAPI

In configuration 1 the main office communicates directly with the interviewers and visa-versa. The main office is responsible for defining the interviewers and distributing the instruments and survey cases amongst them. In configuration 2 the communication between the main office and the interviewers is using a hub created with BLS-Connect. For the rest this configuration is the same as configuration 1. Configuration 3 introduces the use of local agents. In this case the main office only communicates with one or more local agents. These local agents are responsible for defining the associated interviewers and distributing the instruments and survey cases amongst them. The main office in its turn is responsible for defining the local agents and the distribution of the instruments and survey cases to these agents. Similar to configuration 3 is configuration 4. The only difference is that for the communication between the local agent(s) and the associated interviewers the hub is used. There is no configuration defined with the hub BLS-Connect between the main office and a local agent. The BLS-Connect is mainly designed to bridge some kind of Internet protocol with a remote storage protocol (USB-key or memory card). It is supposed that for a local agent at least some kind of Internet protocol is available.

In the case of CN a combination of configurations 3 and 4 was used.

Figure2. The architecture used for CN.



3.4 Data communication (general)

The data communication process can take place between the BLS-Collector and a BLS-Agent, between the BLS-Collector and a BLS-CAPI or between a BLS-Agent and a BLS-CAPI. If BLS-Connect is included in the communication as a hub there are also connections possible between BLS-Collector and BLS-Connect, BLS-Agent and BLS-Connect and BLS-Connect and BLS-CAPI.

These connections are configured according to some kind of “Client-Server” approach. The first one mentioned operates as server, the second one as client. During a data communication process so-called packages are transported: encrypted zip-files containing the objects with data (see also the paragraph Packages and Data objects). The main difference between the role of the server and that of the client is that only the server can delete packages from the intermediate storage devices such as an sFTP-server or a USB-key. The client only copies the packages from and to the device. Because the client identification is part of the name a client can select the proper packages from an intermediate storage. When a package contains the configuration of a new client it is encrypted with a key constructed from the PIN-code of the client and a common (secret) key of the system. This is necessary because the new client does not know its zip-key until the package is unpacked and processed. All the data communication actions are registered in a log. All packages that are sent or received are stored in an archive folder.

The data communication process always consists of 2 consecutive actions: receiving followed by sending. It can be started from the side of the server as well from the side of the client. The (sub)folders involved in the transport of packages from the office (BLS-Collector) to the interviewer (BLS-CAPI), including the communication via a local agent (BLS-Agent) and/or a hub (BLS-Connect), are called “Off2Int”. The (sub)folders involved in the transport in the opposite direction are called “Int2Off”. With the exception of BLS-Connect the applications store all the received and sent packages in an archive folder.

If the server side started up the data communication the following steps are executed:

1. Move all the packages from the “Int2Off” folder on the intermediate storage to the receive folder of the application.
2. Unpack the packages and process the content.
3. Remove the packages from the “Off2Int” subfolder in the send folder of the application if a confirmation of receipt was received from the client.
4. Add the new packages (including receipt confirmations) to be sent to the client to the same “Off2Int” folder.
5. Synchronize the “Off2Int” folder on the intermediate storage in a way that it is a copy of the “Off2Int” folder of the application.

If the client side started up the data communication the following steps are executed:

1. Copy all the packages from the “Off2Int” folder on the intermediate storage to the receive folder of the application.
2. Unpack the packages and process the content.
3. Create the new packages (including receipt confirmations) to be sent to the client in the “Int2Off” subfolder in the send folder of the application.
4. Copy the complete content of this “Int2Off” folder of the application to the “Int2Off” folder on the intermediate storage.

The archive actions are left out in this overview.

3.5 Data communication (tablet)

The Dell tablets have 4 methods of data communication:

- Wireless Internet
- Bluetooth
- SD card reader
- USB port

Because of security demands the data communication with the tablet is restricted. The Bluetooth, the wireless Internet and the SD card reader are disabled in the BIOS. Storage using the USB port is disabled through the registry. Every tablet has an administrator account and a normal user account. Only using the administrator account can the USB port be activated for storage devices by changing a registry value so it is used for the data communication and even then in a very protected way. The data communication process is invoked by the interviewer by clicking on the start button in the application BLS-CAPI on the tablet. It runs a process that opens the USB port in the registry with the following steps:

1. The process is elevated to administrator level by using its credentials which are stored encrypted in the configuration file. The following steps are run “as administrator”.
2. A task is scheduled which will close the USB port at start up to prevent the situation that if the tablet is stopped the hard way by pushing the power button, the USB-port stays open even after a restart. Restarting the tablet the task will close the USB port automatically.
3. Any active Task Manager is stopped preventing the user from interrupting the process.
4. The Task Manager is disabled so it cannot be started again.
5. The auto play function is disabled.
6. A script is started waiting for the event caused by the insertion of a USB key in the USB port.
7. The user is prompted to insert the USB key.

8. The moment the event is received the script detects which drive it is and closes the USB port so no other USB key can be inserted. The USB-key that was inserted stays accessible even if the USB-port is closed.
9. The files intended for the interviewer are copied from the USB key and the files that are to be sent by the interviewer are copied to the USB key. All files are encrypted with a key unique for every interviewer.
10. The USB key is ejected and the user is prompted to take it out of the USB port.
11. The Task Manager and the auto play function are enabled again.
12. The scheduled task is removed.
13. The received files are decrypted with the unique key of the interviewer. Files that are not encrypted or encrypted with the wrong key are removed.

Finally the USB-key has to be delivered to the local administrator to synchronize the data on it using the BLS-Agent application. On the island of Bonaire the key can be delivered by the interviewer personally to the administrator due to the small distances. However, the other 2 islands Saba and Sint-Eustatius are 900 km away. For that reason a hub called BLS-Connect was developed. This hub, running on the laptop of a local coordinator (can be one of the interviewers) only synchronizes the content of a USB-key with the sFTP server in our office (See Figure 2). The local agent can connect to this sFTP server using BLS-Agent to synchronize the data on his side.

The synchronization between the local agent and the application BLS-Collector at the office also uses the sFTP server.

3.6 Data communication (protocols)

All the data communication processes are built as separate scripts and can easily be replaced by other scripts thus changing the data communication protocols. The software on the laptops and tablets used for the data communication consists of Maniplus scripts, (encrypted) VBScripts and freeware.

There are different types of data communication protocols possible. The protocols are defined in the configuration or settings files and using procedures in separate alien MSU-libraries and are therefore easy to replace. At the office side (BLS-Collector) all the available data communication protocols can be enabled at the same time. This can also be done by the local agent (BLS-Agent) when communicating with interviewers (BLS-CAPI). However, the local agent can only select the protocols that are assigned to him by the office. For the communication with the office the local agent can only choose one of the assigned protocols. The interviewer side can use only one protocol at a time. The possible protocols are assigned by the local agent and are a subset of the protocols assigned to the local agent by the office for this purpose. The local coordinator (BLS-Connect) uses a fixed protocol to communicate with the interviewers (key/memory card) and can select one protocol from the assigned protocols to communicate with the local agent or office (depending on who created the local coordinator). The subset is stored in the settings of the local agent/local coordinator/interviewer and can be updated remotely by the party who assigned them.

All the data communication protocols use an in-between medium. This can be a server (e.g. in the case of sFTP or HTTPs) or a remote storage medium like a memory key or card or even a set of folders on the computer in the case of a dummy data communication process for testing purposes. All the in-between media use two separate folders: one for the traffic from the interviewers to the office and one for the traffic from the office to the interviewers. Although the names are configurable and therefore not mandatory it is highly recommended to use the names “Int2Off” and “Off2Int” for these folders.

There are 4 procedures in the alien MSU-library available to support a data communication protocol. These procedures have the fixed names “Preset”, “Receive”, “Send” and “Close”. Not all the procedures are compulsory, this depends on the application. Also the set of parameters can be different for each application.

3.7 Packages and Data objects

During a data communication process so-called packages are transported: encrypted zip-files containing the objects with data. The packages are named “*clientID_dateTime.csp*” (csp=Capi Survey Package) and are encrypted with a 64-character key from the client, applying 256-bits AES encryption. The “*clientID*” is the ID of the client involved in the data exchange. It can be the interviewer or the local agent. In some very special cases it is the ID of the BLS-Connect. Every package contains one or more data objects. Normally these data objects are packed in the root of the packages. Only in the cases where a local agent (with the application BLS-Agent) is sending on data objects from an interviewer (e.g. completed forms) are these data objects packed in a subfolder with the ID of the interviewer. Not only does it avoid conflicts between equal names of data objects coming from different interviewers it also passes on the ID of the interviewer who sent the data object in the first place.

All the data exchanged between the 4 applications is packed in so called data objects. These data objects are zip-files named “*Type{name}dateTime.cso*” (cso=Capi Survey Object). The Type can be “ISM” (= instrument) in case the content is related to an instrument, “SVC” (= survey case) in case the content is related to survey cases or “NTF” (= notification) in all the other cases. The “name” depends on the type of object and its content. The “*dateTime*” contains the date and time of creation and makes the name of the data object unique. Every data object zip-file contains an xml file called “_object.xml” with the properties of the object and, in most of the cases, a zip-file called “Data.zip” with the payload.

Any received object (with the exception of a confirmation object) causes a confirmation message to be sent back. This is done in the same data communication process as that in which the objects were received. At the office side the data communication process includes the updating of the in-between folders based upon these confirmation objects. At the interviewers side no updating is done. If at the interviewer side the same package is received more than once because the in-between folder has not been updated yet by the office side it is skipped based upon the registration in the logging.

3.8 Languages

The display texts of all the 4 applications are stored in separate text files with the same layout. Adding a text file with a new language will be automatically detected by the application and made available for selection. However, only languages that are based upon the same code page as is installed on the computer can be used. To use languages that use other code pages the active code page should be replaced using the standard Windows functionality. Only ANSI based code pages are supported. Up to 9 different languages can be added.

Instruments that are created as a Blaise data model can use other code pages through settings in the data model properties. See the Blaise help file for more information on this subject.

The names of the text files with the language dependent texts are “*xxx.lng*” where “*xxx*” is a 3 character code identifying the language (e.g. ENG , NED, ESP). They are stored in the subfolder “Languages” in the data folder of the application. The first record must contain the name of the language between double quotes (“*name*”). This name will be displayed in the selection list. The rest of the text in the record is optional but should not contain any double quotes. The texts are divided into sections. Every module in an

application has its own section. A section starts with the name of the section (which must be the same as the name of the module) between square brackets (“*[section name]*”). Every subsequent record until a new section contains a text item with the layout “*nnn~text*” in which *nnn* is a number (unique within the section) from 001 till 255. Which numbers are used depends upon the application involved. Empty lines are ignored. Substitution parameters in a text are declared as “*^x^*” where the *x* is an integer starting at 1.

3.9 Tablet enhancements

The BLS-CAPI application is designed to run mainly on tablets although it is no problem to run it on a laptop or notebook. Due to the fact that Maniplus dialogs have fixed sizes the screen resolution should be (set to) 1366x768.

As tablets can be used in landscape or portrait orientation, for some of the Maniplus dialogs 2 versions were created, one for each orientation. Before opening a dialog the orientation is detected with a small VBScript and the right version is activated. If the dialog is already opened and the orientation is changed it is not possible to switch automatically to the other version in Maniplus. Instead an invisible image button is added to the dialog in the background. By clicking on a place where no other control is available this invisible button invokes a procedure that detects the current orientation and activates the right version of the dialog. To prepare Blaise questionnaires for both orientations 2 layout sets have to be created in the Mode Library. They must have the same name but with “landscape” or “portrait” added to the end. The questionnaire will switch automatically to the right layout set thanks to some enhancements in the latest versions of Blaise.

A tablet is mainly managed through its touch screen clicking on buttons or radio and/or checkboxes. However, for filling in text an on screen keyboard is needed. In the dialogs where this is the case the standard on screen keyboard (TabTip.exe) can be opened by a separate button. If a laptop is used the on screen keyboard can be closed with the same button (toggle). These dialogs are designed in a way that the on screen keyboard does not cover the text field when it is opened. During the Tablet Workshop held at CSO Ireland in the beginning of December 2014 UMICH presented a keyboard built entirely in Maniplus which could be a good alternative. Touch screen functionality is replaced by the use of the mouse when a laptop is used.

4. Instruments and Survey cases

The CPLS supports different types of instruments. An instrument is normally a questionnaire to be filled in with the answers given by respondents. In some cases, like the Consumer Price Index, the interviewer fills in the questionnaire based upon his/hers own observations. The instrument can consist of one or more Blaise data models, one or more Maniplus setups or a combination of these two types. There are several options regarding the behavior of the instrument stored in a property file. Data files that are used as e.g. a lookup in an instrument do not have to be a fixed part of the instrument. They can be added (or replaced) later in a separate action. The activation of the audit trail is optional. It is even possible to activate the audit trail for a subset of the survey cases. The results are automatically sent together with the completed statements.

The CPLS also supports different types of survey cases:

- **Person**

If the survey case is a person that person should be interviewed. So if the person does not live at the given address any more it normally leads to a non-response unless the interviewer can locate the person at another address. Only one form per person should be filled in.

- **Address**

If an address is the survey case an interviewer should normally interview more individuals at that address. E.g. in the Labor Force Survey for every household living at that address a questionnaire must be filled in. Another example is a visitor exit survey for which a random set of travelers in an airport has to be interviewed. The airport is then considered as an address.

- **Establishment**

In the case of an establishment (enterprise) being a survey case only that establishment should be interviewed, similar to a person. Who is giving the answers for that establishment is not important. It can be the director, the head of bookkeeping or even an accounting office doing the administration for that establishment. It is also possible that several individuals are giving the answers.

The Labour Force Survey (LFS) is based upon an address sample. That means that the survey cases are addresses but the questionnaire is to collect data from a household. At each address one or more households can be living. All the households living at an address should be interviewed. This implies that the interviewer must be able to add households to an address. To support that a special type of instrument was defined called “multiform”. Activating a form of that type in the BLS-CAPI application causes a dialog to be opened with which the interviewer can add and select households.

Figure3. The “household” dialog from the LFS.

Name	Status
Bolster	opened

The questionnaire involved must have an extra field in the primary key that will receive a sequence number for each added household. The function is designed in such a way that the (multilingual) labels (texts) for this dialog are taken from the property file of the instrument. This makes it possible to use the same functionality also for other types of surveys such as the Visitor Exit Survey.

5. The pilot in The Caribbean Netherlands

One remarkable problem occurred for the LFS on the islands Saba and Sint-Eustatius. As it was an address sample the interviewer should visit addresses and not try to locate certain individuals. Although formally there were addresses they were not known to the population. Also street signs and house numbers were lacking. Finally maps were used to locate the addresses.

In general the interviewers were quite satisfied with the tablets and the application. They had no problem with the size or the weight of the Dell tablets. The contrast of the screen was sufficient. As the interviews were held inside they were not bothered by sunlight. The battery lasted for several days so there was no problem during the interviews with the energy.

Starting up the tablet and the LFS questionnaire was easy although the fact that they had to log on to the tablet first and after that to the application was considered annoying because they had to remember the different numbers and codes. This could lead to the situation where interviewers kept the codes on a note with the tablet. As the protection is regarded as necessary a solution should be found in making it easier to get the interviewer code e.g. by using a dropdown box.

The interviewers had no remarks about the layout of the main dialog as well as the layout of the questionnaire. All the texts were clear, equally well in Dutch as in English, and no buttons were missing. As usual there was no need for context sensitive help. Even the separate manual was hardly used. As there was only one questionnaire available the interviewers mainly used the single address list in the main dialog instead of the separate list for questionnaires and addresses.

Figure4. The opening screen from the LFS.

The screenshot shows the opening screen of the 'Labour force survey Caribbean Netherlands' application. The main text area contains the following content:

The first questions are about your household.

How many persons does the household consist of?

>>Do NOT include children and other family members, who are not registered at this address.<<

Below the text is a large empty rectangular box for input.

On the right side of the screen is a vertical toolbar containing the following elements from top to bottom:

- Navigation buttons: back, forward, and a red 'X' button.
- Navigation buttons: back, forward, and a document icon.
- Numeric keypad: buttons for digits 1 through 9, 0, and a comma/decimal point.
- A button with a keyboard icon.
- Language selection buttons: flags for the Netherlands, the United Kingdom, and Spain.
- A button with a globe icon.

At the bottom of the screen is a status bar with the following text: Old, 1/10/17, Modified by rules, Dirty, Navigate, AKO_ON, Htmain.AantPers.

The On Screen Keyboard was used without any problems. However, the staff expects that with a physical keyboard less typo's will be made. The interviewers normally put the tablet down on a table or their lap during the interview. Because the tablet was placed on a surface there was no need for a special handle to hold it up. Maybe during the upcoming Visitors Exit Survey where the interviewers walk around on the airports a handle could be an improvement. It was decided to lock the tablets in landscape mode as the interviewers were complaining about the automatic rotation. The tablets were too sensitive, so a small movement quickly caused a re-orientation. Besides, they also found the portrait mode less pleasant to use.

The majority of the interviewers used their finger to operate the touch screen instead of the pen. The staff at the office preferred the pen. Although there is a function included in the main dialog to register

appointments the interviewers mainly used their own agenda or notebook. As they don't have to logon to the tablet and the application this worked faster. Also the function of sending and receiving messages was not used. On the island of Bonaire there was often personal contact between the interviewers and the staff so even sending messages was superfluous.

The main problems that occurred were in relation to the data communication. Due to security policies the communication with the tablets was done by USB-key. Starting the synchronization the program prompts for insertion of the USB-key. It automatically detects when this is done and consequently processes the content. However, in the beginning the detection did not work always. It appeared to be caused by the USB-keys. After changing to another type of USB-key the problem was solved. The next problem involved the sFTP connection between the office and the agent. At a certain moment our office in CN reported that for some time they hadn't received any packages from the sFTP server. It appeared that the infrastructure department migrated the server to another environment and that the host key fingerprint was changed without informing us, this due an overload of work for the infrastructure department at that moment. After updating the configuration with the new host key fingerprint the sFTP was still not working. A lot of more errors were made during the migration, all caused by an understaffing of the department. The errors were solved immediately when they were reported and the packages were sent and received again. Later a problem occurred with the sFTP connection with island of Saba. This was caused because the communication was invoked from behind a firewall that blocked the sFTP port 22. The lesson learned was that the software should report any communication problems immediately and store, in case of an error, as much debug information as possible in the logging.

The staff using the applications on the side of the office and the agent were also content with the system. Assigning addresses to the different interviewers was at first reported as difficult but after adding another sorting capability it became much easier. There is no doubt they will continue to use the system for data collection. Of course, based on the first experiences they reported some points of improvement. The next intended surveys will be the Price Level Survey and the Visitors Exit Survey. The Price Level Survey is built in Maniplus. The Visitors Exit Survey will be a Blaise 5 instrument.

6. Further development of the CPLS

As a result of the pilot in CN several small errors were detected and repaired in a newer version of the applications. Also some missing functionality was added.

To be prepared for the future a run-time version of Blaise 5 was created by adding the necessary files (mainly dll's) to BLS-CAPI and creating and starting the Blaise 5 service during installation. To get it running several Blaise 5 configurations files (text but also SQLite) also had to be updated during installation while adding a new instrument. All these functions were added to the latest version of BLS-CAPI. As the rest of the CPLS is still working with Blaise 4.8.4 stored statements were, after closing the interview, on the fly converted to a Blaise 4 data base. It appeared that the Blaise 5 conversion program was too slow so a small C# program was developed using the Blaise 5 API to convert the record from the Blaise 5 bdbx file to an Blaise 5 XML-file. As the XML definition is the same in both versions the record could easily be loaded into a Blaise 4 bdb-file using a Maniplus setup. This solution makes it possible to run Blaise 5 instruments on the tablets or laptops while the office can still process the data using Blaise 4.

Although the CPLS has already proven its added value further use will create needs for new functionality. The final version is not yet there. Already several improvements and extensions have been implemented in newer versions:

- Remote update of languages, configuration and software

- Including survey case related data with survey cases
- Showing the content of received statements
- Improved search and export options for survey cases
- Returning an unused survey case to the office
- And other similar enhancements

Maybe more organizations can join to improve, extend and test the CPLS as a kind of Open Source project.

Figure5. A Blaise 5 form opened in BLS-CAPI.

The screenshot shows a web-based survey form titled "Visitors exit survey" in a browser window. The form has a green header bar with the title and a language dropdown menu set to "English". The main content area is white and contains the following questions:

- How did you arrive on Saba?**
 - ☒ Airplane
 - ☐ Boat
- What was your arrival date?**

03 - 01 - 2015
- So you stayed 3 nights (0 = day trip) on Saba?**

>>If this is NOT right, improve the arrival date.<<

☒ Yes

At the bottom right of the form, there is a green button labeled "Next".

7. Epilogue

I want to thank Rod Furey for reviewing this paper. He improved the spelling and his comments made it more understandable.

Lessons Learned on Preparing and Managing Mixed Mode Surveys

Gina Cheung, and Patricia Maher, University of Michigan, Survey Research Center

Abstract

Preparing for and managing the collection of surveys using multiple modes and ensuring the resulting dataset is of appropriate quality is the challenge faced by all survey organizations. Often the goal is to program one instrument that works both for interviewer-administered and self-administered survey procedures. However, developing one instrument for multi-modes poses nontrivial challenges for questionnaire design (i.e., how the question is asked), instrument design (i.e., visual layout) and programming (i.e., complexity of programming and data processing). In addition there are many challenges related to the technical systems for managing the operational issues of sample and survey management with multiple modes. Multiple modes clearly add complexity to the entire process.

This brief paper outlines some of the challenges and practical approaches taken for handling an integrated approach for multi-mode survey collection. Specifically, we discuss some of the challenges in designing and implementing technical systems to support the complex methodological and operational approaches in these situations.

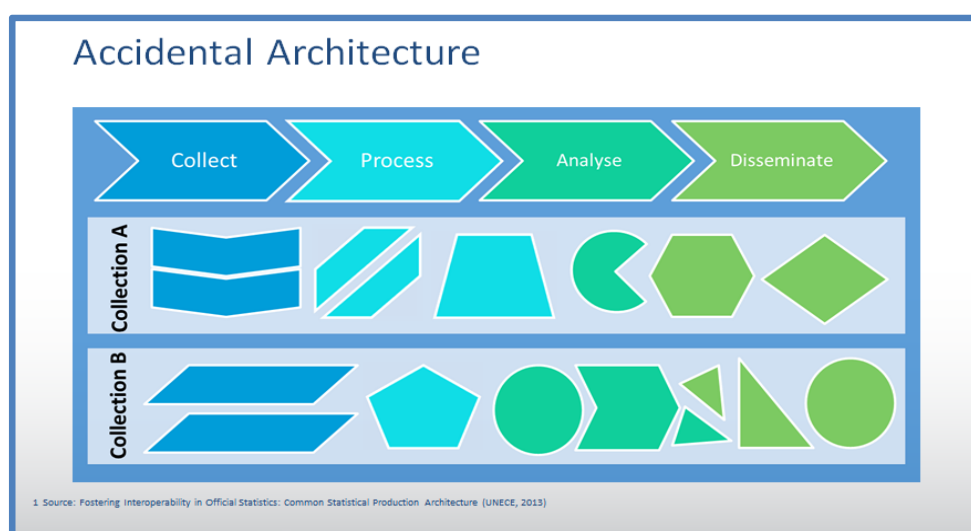
The authors wish to thank Lisa Holland, Director Statistics and Methods, Survey Research Center for her direct input on and review of this paper.

Introduction

More than 10 years ago, Edith de Leeuw (2005) stated that *“One of the most important challenges to survey researchers is deciding which data collection method or mix of methods is optimal...”* Since that time, there has been a steadily increasing literature documenting the need for and advantages of mixed mode data collection. Among other objectives, mixed mode surveys have demonstrated effectiveness in containing costs and in reducing certain types of errors in surveys. A mail survey can significantly reduce coverage error in data collection, and it has been shown repeatedly that an interviewer-administered survey is expected to obtain a higher response rate than a self-administered survey (Dillman, de Leeuw). Many major survey programs – the American Community Survey (U.S.), the Health and Retirement Study (U.S.), Labor Force Survey (UK), and the European Social Survey – have used or experimented with using and testing mixed mode data collection designs.

As the increased level of effort to effectively contact and engage respondents in survey requests continues to put pressure on research organizations to find more efficient methods for collecting complex human measures, we quickly learn that less well documented are the implications of mixed mode designs on the technical systems required to carry out these projects. Implementation of mixed mode design requires not only early planning and careful consideration of both operational and methodological design decisions in order to properly manage data collection protocols in production but also a thoughtful design to report and evaluate the protocol effectiveness during collection. The push to create integrated measurement protocols -- surveys collected by phone, in-person or self-reports on the web -- create unique challenges for survey managers, software developers, data managers, and IT professionals. Many survey organizations began collecting mixed mode surveys without a full set of technical tools, or an integrated solution. This ‘accidental architecture’ as labeled by the United Nations Economic Commission for Europe (UNECE, 2013), and further described recently by US Census Bureau (Thieme and Mathur, 2014) ultimately proves to be insufficient for sustainable collection procedures with complex surveys across multiple modes.

Illustration 1: Accidental Architecture



This short paper describes some of the primary design issues associated with mixed mode survey questionnaires and the implications of those design features on technical systems. We also end with a short summary of the newest integrated survey management system used by the University of Michigan, Survey Research Center, for multi-mode data collection projects.

Study Design Considerations

There are two areas of study design that we will address separately. The first is the overall data collection protocol, and the second is the delivery of the sample. The data collection protocol is a complex set of decisions that are ideally made prior to the development of any technical systems. The most important decision is probably the modes in which the survey will be available to the respondent, but even that decision has numerous design questions which need to be answered:

- What mode will be used to contact the respondent?
- Will contact attempts be made in multiple modes?
- Will the mode be assigned to the sample unit or will the respondent have a choice?
- What order will the modes be available to the respondent?
- If a less expensive mode is used first, when will the switch be made to the other mode?
- After the switch, would the respondent still have the option of completing the questionnaire in the first mode?

These basic decisions will drive the specification for the technical system. The more options available to the respondent, the more complex the specification. As surveys continue to implement more sophisticated techniques for cost management and the reduction of errors, technical systems are pushed further to manage these complexities. The use of propensity models, adaptive designs, or sub-sampling allow for more controlled follow up of sample for reduced costs or errors, but in turn, add complexity to the technical requirements and demands on the technical solution implemented for the survey. Finally, the supervisory structure and size of the field staff may also require unique features in the technical systems.

Delivery of the sample is the second area to consider in study design. Ideally a technical system would be flexible enough to handle sample in various formats and from multiple frames. The advantages of mixed mode designs may result in more complex sample designs, for example a combination of address-based sample and telephone-based sample. A panel study using a list sample may elect to add a cohort from an address-based screening initiative. The technical system must be able to integrate these samples, keeping track of the mode of data collection, and ideally be able to report on them both in aggregate and separately and in timely manner. Additional questions related to sample delivery include:

- Will there be multiple sample releases or sample replicates?
- How are cases assigned to interviewers?
- Does the interviewer need the capability of transferring a case?
- If the respondent makes a request, can the interviewer assign the case to another mode?
- Does a case need to be available in concurrent modes (e.g., web and CATI)?

These are only a subset of the design questions that need to be answered prior to the specification of technical systems. Design features such as the use of respondent incentives, inclusion of biomarkers, refusal aversion protocols, the need for respondent signatures, or any other design feature will have an impact on the management system.

Instrument Development Considerations

Instrument development in multiple modes has a unique set of challenges and perspectives. There are certain design issues that are unique to uni-mode development, that is, trying to design a single data collection instrument that can be used in multiple modes. The issues are different, however, if a researcher is attempting to adapt an existing questionnaire from one mode to another. Furthermore, there are varying perspectives on the objectives of mixed mode design. Some approach this issue with the view that questions should be kept as similar as possible across modes, supporting the principle of standardization that all respondents should be presented with an identical stimulus. Others believe that it is better to optimize each mode, writing questions specific to that mode, with the goal of collecting comparable data across modes.

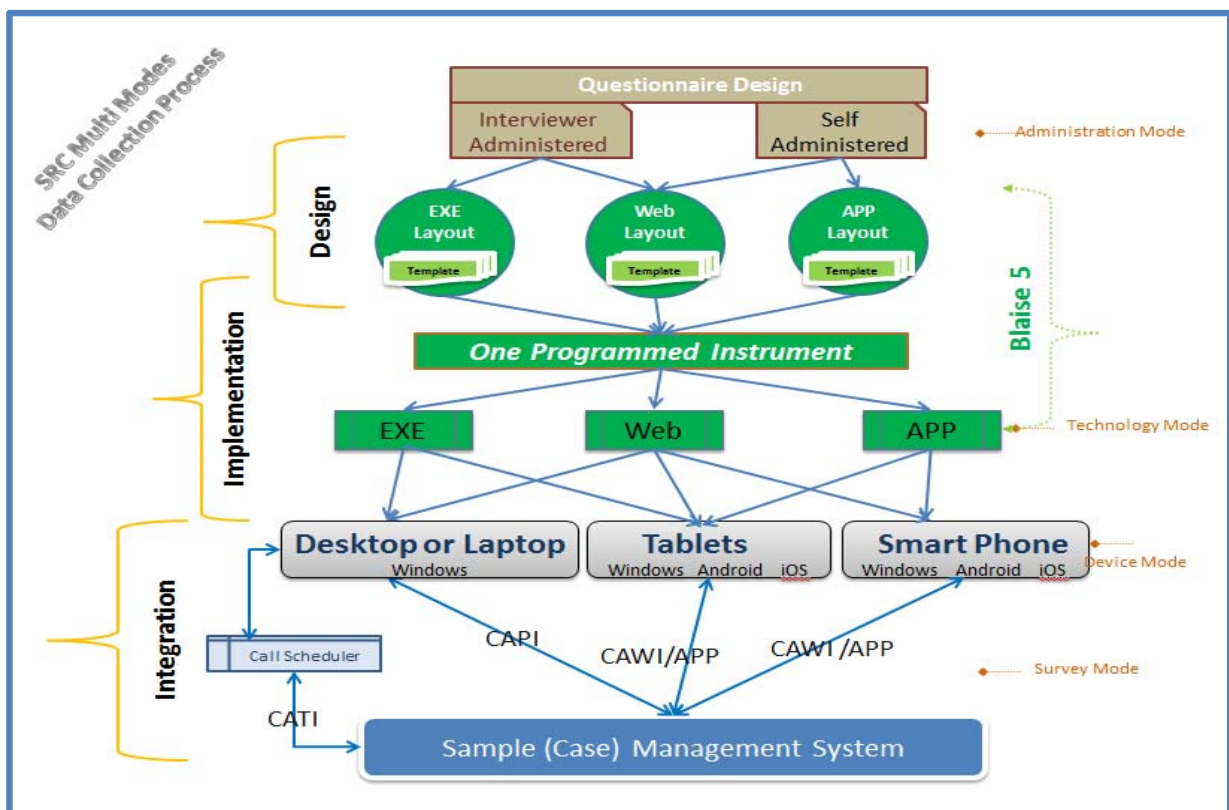
Here we will provide an overview of multi-mode instrument design from an IT perspective, we will briefly discuss questionnaire design considerations and the use of various devices for completing self-administered surveys, and again, we will focus on how these complexities impact the requirements for technical management systems.

Looking at the overall development and implementation required to support a multi-mode data collection protocol, again we are struck by the complexity, and particularly by how much complexity is introduced at the instrument development stage. Questionnaire design, instrument testing, device testing, reporting, and integration with the sample management system all create demands on the software and systems we rely on for implementation. Illustration 2 attempts to expose at least four main focal areas:

1. Questionnaire or instrument design – There is a basic decision to make about whether data collection will be self-administered, interviewer-administered, or both. Question wording and layout decisions will depend upon this decision.

2. Programming the instrument(s) and the technical approach for delivery – Software selection and the decision about whether to create one instrument or multiple instruments will be critical information to exchange seamlessly with the sample management system.
3. Integration of the various hardware environments – This involves working within an existing hardware infrastructure, or may involve hardware selection decisions which are still within the control of the researcher. For web surveys, however, there are greater considerations related to hardware as the researcher does not have control over the device with which a respondent may decide to complete a questionnaire.
4. Sample management functions – That is how to facilitate the data exchanges and output for reporting. Reporting requirements can be extremely complex and possibly simplified by using a data warehouse approach. (This paper does not address this topic.)

Illustration 2: Multi-Mode Data Collection Processes with IT Lens



Questionnaire Design Issues

As mentioned earlier, questionnaire design considerations may differ depending on whether the objective is to adapt an existing instrument into a new mode or to develop a new instrument to be used across modes. In either situation, however, each survey question must be evaluated to determine if it will work across modes, if it can be adapted to fit both modes, or if it does indeed need two different designs and/or layouts. We provide examples of only a few of the questionnaire design considerations that we face most frequently; we discuss these in the context of adapting an interviewer-administered questionnaire for self-completion in the web mode.

- **Interviewer instructions** – The researcher needs to make a decision about how to handle instructions that were intended only for the interviewer. These may include how to respond if a respondent volunteers an answer outside the code frame, or how to “field code” what appears to the respondent as an open-ended question. With the knowledge we have from web surveys about how much information can be presented on the screen, these issues can pose significant challenges, especially when the researcher is trying to maintain the comparability of data across modes.
- **Help text/definitions** – The literature on web surveys (Couper, 2008) also informs us that respondents are unlikely to seek help or click on definitions, even when they are made available in web surveys. This poses the question of whether to present this information to respondents. We equip the interviewer with these resources if necessary, but adding the detail to every question would likely have an adverse effect on survey length, respondent burden, and completion rates.
- **Optional text (usually in parenthesis)** – Many interviewer-administered surveys include text that does not have to be read to the respondent unless needed. This may include the repetition of a reference period, a stem question, or a response code frame that is being used repeatedly. The researcher must decide how best to have this information available to the respondent at any point in a web survey without increasing respondent burden.
- **Item missing data** – A common struggle in multi-mode questionnaire design relates to whether or not to present “Don’t Know” and “Refuse” options. In interviewer-administered surveys, it is common to record these answers when volunteered by the respondent, but not to offer them explicitly, because offering them is likely to increase the likelihood that they are selected (thereby increasing the rate of item-missing data). In a web survey, we do not want to offer the categories for the same reason, but without them, it is impossible to know whether a respondent skipped a question intentionally or inadvertently.
- **Grids** -- There are instances where a grid design increases efficiency for interviewers and for respondents, but the design, layout, and length of grids needs to be considered carefully within the context of the survey, and the devices on which the survey might be completed.

These decisions affect not only the complexity required of the technical systems, but also the resulting data quality. In the case of item-missing data, it is not difficult to measure the mode-effect being introduced by a multi-mode design (the technical system should be designed to capture and report this information), but that can be more difficult to assess in other situations. An invaluable feature of survey software is the ability to designate the mode of administration and call out to the appropriate layout when the question is administered by an interviewer and when it is self-administered. However, even with this feature, there are still constraints on question design and layouts due to the underlying data structure.

There are additional design considerations that affect both the programming needs and the data quality. One example is the navigation of gate questions. Gate questions are questions that guide significant substantive paths through a questionnaire, and they are critical to the data. The researcher can decide to “prompt” a respondent to complete a gate question that is skipped in a self-administered survey, but this option needs to be used with discretion so as not to annoy the respondent, and lead to increased break-offs from the survey. Other items related to navigation include whether or not a respondent is allowed to back up in a questionnaire, and if so, are any previous responses retained or deleted as the respondent moves forward again.

One of the most common issues in mixed mode design discussions these days relates to the device with which a respondent chooses to complete a self-administered questionnaire. Desktops, laptops, tablets, and smart phones each present a new consideration for the designer, and the complexities are multiplied as different browsers and operating systems are introduced. These growing options for respondents have costly implications for design, programming, testing and integration of survey instruments. Researchers may decide to limit respondent options in this regard, but they are likely to introduce other errors if they make this choice.

Even after the questionnaire design decisions are made, there are still decisions about how best to implement these decisions. The technical team will need to work collaboratively with data managers and programming staff to consider whether it is optimal to create separate or integrated instruments. This will depend on the length and complexity of instruments, how many design decisions vary by mode, and how difficult the data harmonization process will be. Taking the underlying data architecture into account in the design phase will facilitate reporting capabilities and data management on the back end. It is also critical to consider what paradata are being collected and how those data are being used.

Mixed mode designs have greater implications regarding technical infrastructure as well. Web survey administration requires immediate communication with servers where CAPI administration does not. Web surveys have further considerations related to the speed and performance of the instrument which, again, expands the testing task as options for devices and operating systems increase. If the respondent is allowed to switch between modes, that is, to start the survey in one mode and complete the survey in another, additional features need to be added and tested. This is also true if a survey is using auxiliary systems for data collection, calling out to other programs for additional data collection.

All of these systems require the integration of features for managing the various components of the data collection design with the features required to manage and monitor the performance of the sample. Furthermore, reporting features must be integrated and tested in order to provide the management tools that will be critical as soon as production begins.

Technical System Considerations

It is clear that every design decision has implications for the technical systems required to manage a project, and that the complexity increases exponentially as features are added, developed, tested, and integrated with one another. What poses an even greater challenge is the creation of a system that is flexible enough to accommodate multiple projects with varying levels of complexity. The remainder of this paper focuses on developmental work being carried out at the University of Michigan Survey Research Center for the purposes of survey and sample management. We talk first about information management needs, other system requirements and proposed solutions. Then we will describe our work related to the development on the Michigan Survey Management System (MSMS) and some of its specific capacities.

An integrated management information system for handling mixed mode data collection projects must include features for both sample management and survey management. The sample management functions must include control of preload data, sample releases, sample replicates, mode, interviewer assignment, case transfer, and closing cases. Integrated systems are also required to manage all process functions related to the data collection including prenotification, respondent communication, administration of the survey itself, other study protocols (e.g., biomarkers, neurocognitive testing) logging receipt of materials, and the status of these activities at both the case level and the task level. Finally the systems must work together to maintain reporting capacity that is flexible to enough to monitor production rates in aggregate or broken down by any of the subgroups mentioned above.

System Requirements and Solutions

In order to execute all of the functions that have been described thus far, the following features are desired:

- The system should be able to support mixed- and multi-mode protocols in one project. This terminology may not be universally understood, but the issue speaks to the capacity for a project to have cases assigned to given modes and to use multiple modes within a single case; the protocol can include sequential or concurrent modes.
- All protocols should be automated to the greatest degree possible so that they can be repeated and applied consistently across respondents, interviewers, iterations of reports, etc.
- Survey management and sample management capabilities should be integrated into a single system.

- The system should accurately reflect survey management process, that is, the steps required to complete a successful data collection should be apparent to the user.
- Ideally, a well-integrated system would ultimately reduce the effort required to collect data, resulting in efficiencies in either time or cost, or both.
- Minimize application recompiles and deployments.

As we work toward the full development and implementation of this system, we have found the following solutions to be effective in meeting our objectives:

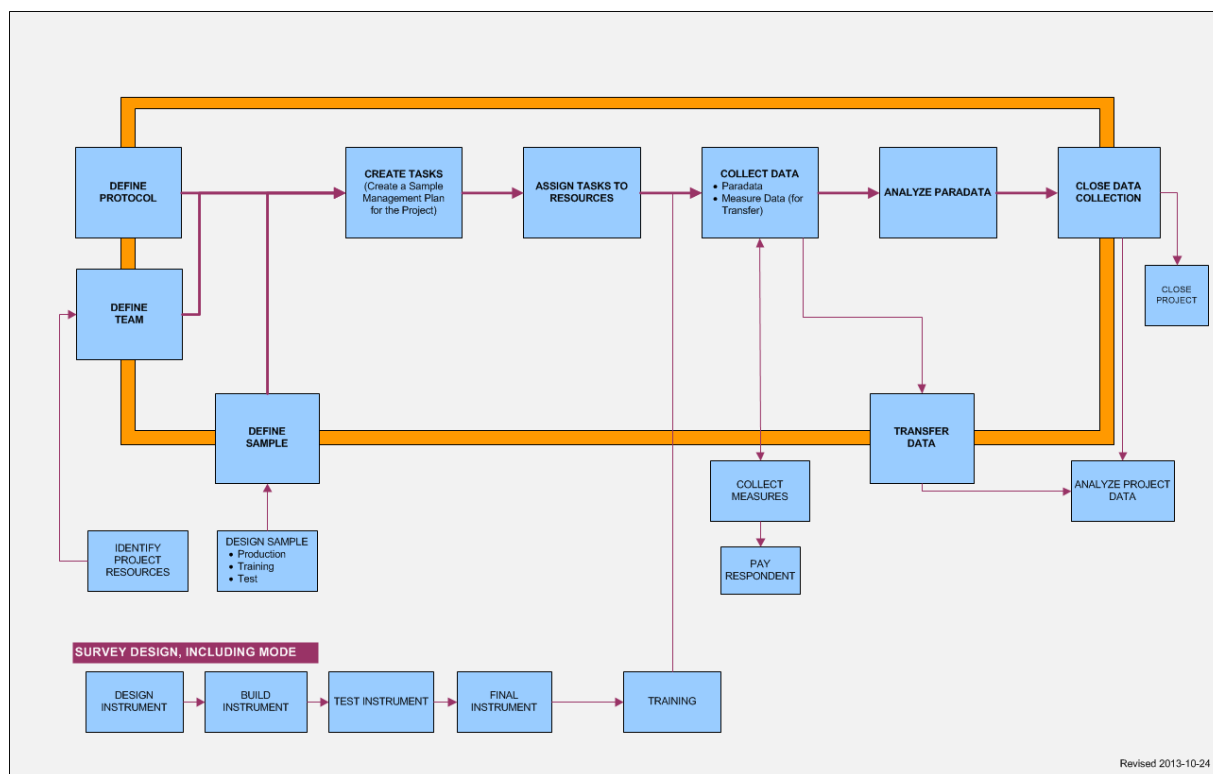
- We have implemented automated rules sets that can be applied to important process steps, e.g. task rules.
- We have adapted a task-level structure throughout system.
- The system is characterized by its interrelated set of applications and services.
- The system architecture reflects business objects and their relationships to one another.
- Project implementation is operationalized through configuration files instead of through changes to compiled code.
- Project-level configuration is intended to provide a balance of standardized and customized features.

Michigan Survey Management System (MSMS)

The Michigan Survey Management System (MSMS) is an integrated survey management information system for handling mixed mode data collection projects, across all users – respondent self-administered, interviewers (telephone or field) and external collection activities such as verification call backs or managing follow-up contacts (e.g., scheduling phlebotomy appointment, etc.). MSMS reflects a change from using a sample-based framework, or case “ownership”, to a task-based system which allows for greater flexibility of task assignments across users. It provides users with concurrent access to sample information such as call records and contact history (capturing both in-bound and out-bound contact attempts) rather than limiting that information to the interviewer to whom a case was assigned. As an example, the system can be used to sequentially send an advance prenotification letter to the respondent, send an email invitation to participate in a web survey, prompt the interviewer to make a non-response reminder call, send a reminder email message and deploy the survey instrument in either the CATI or web mode.

The illustration below reflects basic survey design and flow. The thick orange line defines the boundaries of MSMS and external processes or systems. The survey process was evaluated closely to inform this new IT management system.

Illustration 3: Survey Process Diagram



MSMS Customization

The Survey Research Center routinely conducts multiple projects concurrently, and we are likely to have as many unique designs as we have projects in production. It is important to be able to generate standard metrics to monitor progress that work across projects, but it is equally important to be able to tailor technical systems to meet the unique requirements specified in any data collection protocol. We have worked hard to achieve the optimal balance of these two needs. The MSMS configuration allows each project to use and show the attributes appropriate to its protocol. To support an environment of continuous improvement, the system has been created such that protocols can leverage and improve upon what has worked well, without affecting other projects. MSMS applications have designated regions within which projects can define the content that best suits their needs. This definition is done using configuration files that are changeable throughout the project without redeploying the application.

Finally, to meet the needs of survey researchers today, it is essential that technical systems be developed to provide access to process data (paradata). Examination of these data can inform decisions and guide the trade-offs during the data collection, but for this to be effective, the paradata need to be available to the research in a timely manner. The development of a mixed mode technical management system should facilitate access to paradata that might inform production decisions such as when to switch modes or other aspects of responsive design. This might include access to actual survey measures, as it may be of interest to examine the effects of mode on survey estimates, and it may include access to process data in addition to the more traditional key stroke files that come to mind when we think of paradata.

Closing Summary

Robust data capture software systems which provide options for multi-mode, cross platforms and devices, self-administration and interviewer- administration provide great opportunity for complex survey and ancillary data collection. As we continue to develop and test these systems, it is useful to embed small experiments where possible to further inform the development of the systems as well as the data collection protocols themselves. It is also important to emphasize the need for built-in flexibility as the increasing growth in technology will continue to present us with both challenges and opportunities.

REFERENCES

Couper, MP. Designing Effective Web Surveys. Cambridge University Press, 2008.

Dillman, D.A., Jolene D. Smyth, and Leah Melani Christian. Internet, Phone, Mail, and Mixed mode Surveys: The Tailored Design Method 4th Edition. Wiley, 2014.

de Leeuw, E. D. "To Mix or Not to Mix Data Collection Modes in Surveys," *Journal of Official Statistics*, Vol. 21, No. 2, pp. 233–255, 2005.

de Leeuw, E.D. and J.J Hox. *Internet Surveys as Part of a Mixed Mode Design*. Pp. 45-76. in Social and Behavioral Research and the Internet. Ed. M. Das, P. Ester & L. Kaczmirek. New York: Routledge, 2011.

Martin, P. and Peter Lynn, P. "The Effect of Mixed Mode Survey Designs on Simple and Complex Analyses." Centre for Comparative Social Surveys, City University London. Working Paper No. 04, 2011.

Thieme, M.T. and Anup Mathur. "Designing and Architecting a Shared Platform for Adaptive Data Collection in Surveys and Censuses. U.S. Census Bureau (2014)

United Nations Economic Commission for Europe (UNECE). "Common Statistical Production Architecture." Version 1.0, pp. 4-5, 2013.

Contact Information

Gina Cheung qianyang@umich.edu

Patricia Maher pmaher@umich.edu

Deploying an Automated Data Read In Process at National Agricultural Statistics Service

Roger Schou and Emily Caron, National Agricultural Statistics Service, USA

1. Introduction

The National Agricultural Statistics Service (NASS) uses multi-mode data collection including CATI, electronic data reporting (EDR, also known as CAWI), CAPI, and paper data collection. Blaise is best known as our primary data collection source for CATI, but it is also one of the primary editing tools in NASS. The data from all other modes of collection need to make it into Blaise for editing. For many years, reading data into Blaise involved users manually kicking off a Maniplus program and then sitting there to watch it spin.

In 2010, we changed the way in which we stored our Blaise data. We moved from individual decentralized Blaise data sets in 42 separate locations to one centralized MySQL transactional database accessed by all states. All of our surveys' data are stored in the same set of tables in MySQL. We chose in-depth generic data storage so that the back end process which reads from the MySQL database and writes to a RedBrick analytical database could be one standard ETL (Extract-Transfer-Load) for all surveys.

As more and more surveys began using our new Centralized Blaise solution, the number of times the individual data read in processes ran also increased. With all of these users accessing the same resources to get data from the non-Blaise systems read into Blaise for editing, the performance quickly began to deteriorate. In May 2014 after realizing what was causing the problem, we redesigned the data read in process to run in a more automated fashion. This new process has greatly improved performance and user experience on many levels.

2. A Problem Arises

In the past when we were decentralized with 42 separate offices, reading data in from our non-Blaise sources was not an issue performance-wise. Each survey had its own proprietary Blaise database containing only records that belonged to that office. Everything was sitting on their Local Area Network, so the demand on resources was not very high. Users would initiate the read in process for that location and watch the process run from start to finish. Included in the read in process has always been an integral check (or batch edit) using the CHECKRULES method in Manipula. Even with this extra piece, the entire read in process was pretty fast since the data didn't have far to travel and not many users were competing for the same resources.

Once we centralized the data storage to a single MySQL database, we introduced the Wide Area Network (WAN) into our processes. We also began moving more and more surveys into the centralized world. The data read in functioned very similar to how it worked in the decentralized world: The user initiated the read in, sat there and waited while data was read into the centralized dataset and batch edited. Performance was not as good as it was with the decentralized environment, but it was still tolerable.

When some of our major surveys with very large sample sizes near 100,000 entered the picture, performance deteriorated to intolerable levels. Some locations were seeing read in times of nearly an hour for a batch of records. It even got to a point where we were assigning time slots for certain offices to run the read in process, since we feared concurrent read in jobs were causing problems.

The NASS infrastructure concerning Blaise changed as the number of surveys grew. In April 2014 we expanded from one to three Blaise Data Servers, splitting our surveys across the three servers. Any

given survey was housed entirely on one given server. This expansion eased the stress that we were once imposing on a single Blaise Data Server, but unfortunately we continued to experience unacceptable lags in the read in process. With some of the surveys having a short two-week turnaround from start to finish, something had to be done.

3. A Solution Is Formed

After doing some testing, we discovered that if we ran the read in process on the Blaise data server rather than from the client machine, it was significantly faster. This basically eliminated the WAN communication traffic. However, we were still thinking in a decentralized manner. We needed to step back and attack this from a centralized perspective.

Testing had also revealed that concurrent read in jobs run for the same survey were degrading performance. We decided it would be best to do a little footwork on the front end to combine all data needing to be read in from the various sources and numerous locations, and then WE would control the Manipula to read in the data and batch edit at a national level. This would eliminate concurrent read in jobs per survey, and we could also take advantage of overnight hours when no users were in the system.

4. Queueing the Data

CATI data is collected using the Blaise instrument, so it is available for editing the instant the interviewer completes the interview and exits the form. Our CAPI and EDR/CAWI data are collected in the same in-house system, so data is manually pulled for both of these at the same time and then read into Blaise. Paper data is keyed in a software called Viking before it's read into Blaise for editing.

We created a *Queue Files for Read In* process which users can run from a button on our VB.NET CASIC Menu when a batch of data is ready to be read into Blaise, whether it's EDR, CAPI, paper, or any combination of the three. This approach allows multiple users to stage their data into the queue, but the actual loading of the data into the database is done nationally. It runs at set times throughout the day, and runs completely on the server.

Users have two options for queueing data: daytime or overnight. The daytime read in process runs every fifteen minutes during the day from 6:00 am through 11:00 pm, seven days a week. The overnight read in runs at 12:15 am, Tuesday through Sunday. When a user clicks the Queue button, as long as the files are named correctly and sitting in a pre-determined location on a designated local drive, it will pick up and move the file(s) to the centralized Blaise Data Server. The file(s) will be renamed to include a date and time stamp and placed in either the QueueDay or QueueNight folder. These folders are located at the survey level on a given Blaise Data Server (i.e. \\KCBLZxx\\casic\\data\\casic\\surveyfolder\\hq\\QueueDay\\, where xx is the server number and surveyfolder is the folder name of a given survey). By adding the date and time stamp, the files are unique (to the second) which almost eliminates the possibility of someone's file getting overwritten by someone else's. It also allows for any number of files to be queued between program runs.

When a user queues data, the program will look for the existence of a "SurveyInfo.txt" file for that particular survey. This file contains pertinent metadata specific to each survey that the read in program will use, like instrument name, whether or not the survey allows new adds, etc. If the file doesn't exist or is older than today's date, the program will generate a new one.

A trigger file is also created during the queue up process. Its file name is made up of the name of the survey folder, the data source, and an extension of .TRG. It is placed in either the TriggerDay or TriggerNight folder, and the file name alerts the sweeping program as to which surveys to process and from what source they originated. The contents of this file contain the command line options to be

passed into Manipula. Multiple trigger files may be present, each representing a different survey and different data sources within surveys. The trigger folders are located at the server level (i.e. \\KCBLZxx\casic\data\casic\TriggerDay\).

5. Sweeping the Data

A VB.NET console application was written to accomplish the automatic data sweep. It is set up as a scheduled task on the Blaise Data Servers. By looking at the files present in the appropriate trigger folder, it will first move the trigger file into a TriggerStage folder and then automatically sweep the corresponding surveys' queue folder. To make this process even more efficient, we used threads to kick off multiple survey read in jobs at once. As long as data wasn't being read in concurrently for the same survey, performance wasn't degraded.

Since the application runs every fifteen minutes during the daytime and some large survey batches may take longer than fifteen minutes to process, it first checks for a file in the surveys' QueueStage folder called InUse.TXT. If it does already exist, that tells us the previous read in for that survey is still running so the queued data will be postponed until the next sweep. If the file does not exist, it creates it there for the surveys that have a trigger file present and the process continues to run.

The VB.NET application concatenates all instances of the EDR/CAPI data into one file and all instances of the paper data into another file for each survey and places them in the QueueStage folder. The TriggerStage and QueueStage folders are used to prevent interference of a user trying to queue data while the sweeping process is active. The program will still allow the user to queue data while the read in process is running, but that data will not be picked up until the next sweep.

Once the data has been concatenated, the console application executes the Manipula setup to read in the data and run integral check. If a record is already complete in the MySQL Blaise tables and the read in program attempts to read in the same record from a non-Blaise source, the system will detect the potential overlay and will write it to a proprietary Blaise dataset that sits off to the side. We then have a Manipus program that will display these potential overlays next to the completed record in MySQL, showing a few important fields. The user then has the option to open either the overlay record or the original record to view the data. There are buttons that allow the user to keep the version they want, and the other one is deleted.

6. Log Files

The automated read in process appends to two separate log files so that we can ensure all is running smoothly behind the scenes, and if it's not we'll have a clue as to where the issue occurred and why.

One log is located at the server level. It creates log entries each time the auto read in scheduled task begins and ends with associated date and time stamps. If there are no trigger files present, then it states that fact in the log file. If there are trigger files, log entries are created showing each trigger file being moved to the staging area as well as the file size of the data file(s) to be processed. It also stamps the start time on the end of each line in parentheses, so it is easier to match up starts with endings in case any overlap each other. Any errors occurring at the high/server level of the program would be seen here. An example of some server level read in log entries can be seen in Image 1.

Image 1. Server Level Read In Log Example

2/2/2015 10:00:00 PM....	START of Read In process. (220000)
2/2/2015 10:00:00 PM....	END of Read In process - no triggers found for DAY read in. (220000)
2/2/2015 11:15:00 PM....	START of Read In process. (231500)
2/2/2015 11:15:00 PM....	BEEPDI140000edr trigger moved to staging area. (231500)
2/2/2015 11:15:00 PM....	BEEPDI140000raw trigger moved to staging area. (231500)
	BEEPDI140000 total data size = 45121 bytes
2/2/2015 11:15:00 PM....	FLMPC150100raw trigger moved to staging area. (231500)
	FLMPC150100 total data size = 577 bytes
2/2/2015 11:15:21 PM....	END of Read In process. (231500)

The second type of log file tracks the process at the survey level, so these logs are stored and accessed under survey specific directories. There are only entries in this log file when there are data files present for that particular survey. Very similar information is displayed in this file when there is data to be processed: Start and end time of the read in process for each type of source file, along with data size. If an error occurs at the survey level of the read in program, it will appear in this log in significant detail with a related log entry also written to the server level log with much less detail.

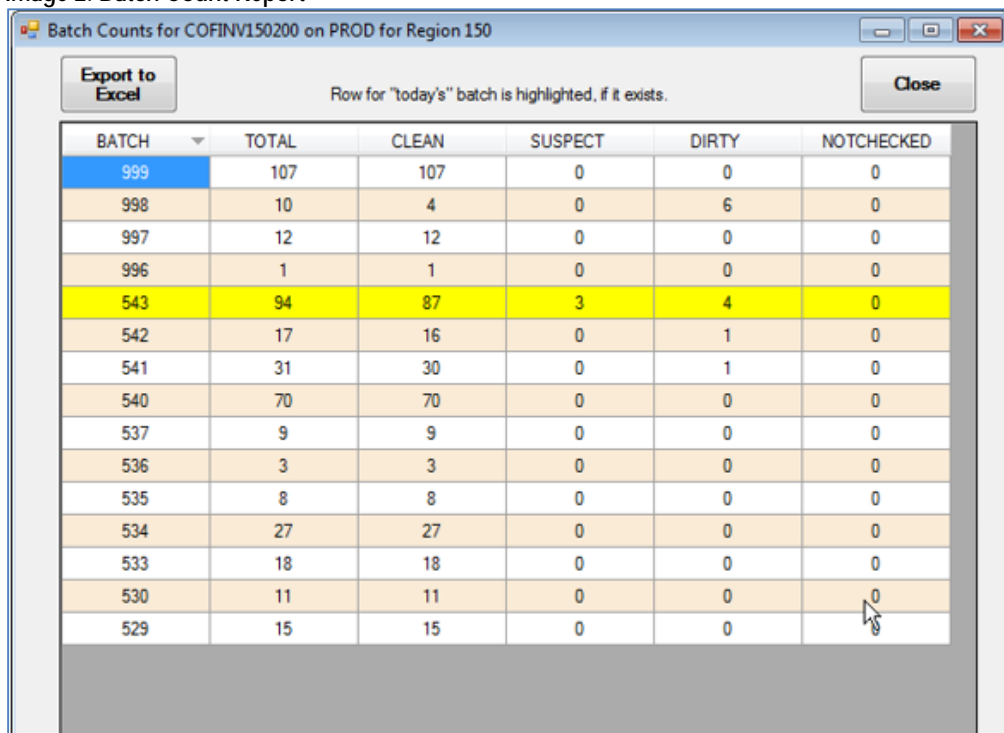
We have created buttons for staff in headquarters to review these log files. When the button is clicked, a copy of the log file is written to their personal temporary folder, so that the log files always remain available to the automated process. This was done to prevent the process from failing due to an inability to write to a file that is in use by someone reviewing it.

7. Improvement 1 - Batch Count Report

We use Blaise to interactively edit our data on many surveys, and the users prefer to edit them by batch. The CATI data is stored in batch numbers equal to the Julian date on which they were collected (1-366). For the auto read in, we decided to assign batch numbers by adding 500 to the Julian date for the day they were read in (501-866).

Even though the auto read in process runs every fifteen minutes, the users were concerned as to how they would know a batch had been read in. They were used to having total control of the read in process, watching it from start to finish and even assigning a batch number for each batch read in. In order to address this concern, we created a Batch Count Report as shown in Image 2.

Image 2. Batch Count Report



BATCH	TOTAL	CLEAN	SUSPECT	DIRTY	NOTCHECKED
999	107	107	0	0	0
998	10	4	0	6	0
997	12	12	0	0	0
996	1	1	0	0	0
543	94	87	3	4	0
542	17	16	0	1	0
541	31	30	0	1	0
540	70	70	0	0	0
537	9	9	0	0	0
536	3	3	0	0	0
535	8	8	0	0	0
534	27	27	0	0	0
533	18	18	0	0	0
530	11	11	0	0	0
529	15	15	0	0	0

This report is created in VB.NET by using a select statement against the MySQL tables. It lets the user know if a batch exists and the number of records within that batch, as well as the cleanliness of the records in the batch. Since most people do not know the Julian date for a given day, the current day's batch is highlighted if it exists, as shown in the image above. Also we added Julian date to our main menu screen for quick and easy user reference.

8. Improvement 2 – Record Level Rejects and Check Ins

Now that the read in program was taken out of the users' hands, we struggled with what to do with a report we used to generate out of the old read in program which alerted users to record level rejects. These rejects occur on surveys that do not allow new records to be added to the sample. For these types of surveys we have a set sample master and if a record that doesn't exist on the sample master attempts to be read in, the read in program will reject it. Often these errors are attributed to the primary key getting keyed incorrectly from paper forms, so the users would want to know about these errors as soon as possible to resolve the matter and get the correct record's data read in.

Another problem had us temporarily scratching our heads. When users were able to read in data at will the old way, this also accomplished records receiving a status of 'complete' in Blaise in a timely manner, which automatically removed the records from the CATI call scheduler. We do have other methods of checking non-Blaise records into Blaise to ensure they won't be called, but users worried about the rare cases when records had been missed being checked in via the other methods. If a user queued up data for the overnight run and it happened to get missed for check in, it could take hours for the record to be read into Blaise, therefore allowing the possibility for it to be called via CATI.

Both of these problems were solved by adding extra code to the queue data step. As users queue up their data files, a Manipula setup is run to check for non-matches between the data files and the sample master. If any are found, they are written to an output file which is presented to the user at the end of the queue up process. Another Manipula runs to ensure all records being queued up for read in are checked into Blaise, thus immediately removing them from the CATI Call Scheduler.

9. Improvement 3 – Item Level Rejects

In order to read in the data from the non-Blaise sources, we use a universal data layout with primary key information as well as an item code and the data. There is always a possibility that this item level data can be rejected.

In the past when users ran and watched the data read in, if item level rejects occurred the read in program would output a text file for them to view which contained this information. However, with the new automated process there was no one watching it run, and each run could contain any number of different users' batches. So how could we alert users about item level rejects?

To address this in our new automated process, we created two fields in every Blaise instrument. One is an OPEN field that the auto read in process writes messages to when it encounters an invalid or blank item code, a duplicate item code, or an item code with a blank value. The second field is an enumerated field with one value for "Reviewed." There is a critical error check in every Blaise instrument that says if the enumerated field is not equal to "Reviewed", then the OPEN field must be empty. This forces the editors to review any bad item code data that failed to read in to the data set.

A text file is also created that lists all rejected data so that a user in headquarters may review them to potentially detect a systematic problem. The text file is a bit cryptic. We realize there is room for improvement on this file, as the headquarters staff are currently relying on the editors dealing with the critical errors to determine if there is a systematic problem.

10. Improvement 4 - Interactive Edit Sort Options

The interactive edit interface at NASS is written in Manipulus. It allows the user to search the database on batch number. It presents a dialog box as shown in Image 3, where it shows the number of forms within each batch as well as the status totals for the batches. The Batch Status takes into account the "dirtiest" status of any form within each batch. From this dialog box, the user can select the batch they would like to edit.

Image 3. Interactive Edit Batch Listing

Select a Batch								
Batch	Number of Forms	Batch Status	Notchecked	Dirty	Suspect	Clean	Sent	
529	15	CLEAN	0	0	0	15	0	
530	11	CLEAN	0	0	0	11	0	
533	18	CLEAN	0	0	0	18	0	
534	27	CLEAN	0	0	0	27	0	
535	8	CLEAN	0	0	0	8	0	
536	3	CLEAN	0	0	0	3	0	
537	9	CLEAN	0	0	0	9	0	
540	70	CLEAN	0	0	0	70	0	
541	31	CLEAN	0	0	0	31	0	
542	17	CLEAN	0	0	0	17	0	
543	166	DIRTY	0	35	3	128	0	
996	1	CLEAN	0	0	0	1	0	
997	12	CLEAN	0	0	0	12	0	
998	10	CLEAN	0	0	0	10	0	
999	107	CLEAN	0	0	0	107	0	

11:15

Accept Cancel

Once a batch is selected, another dialog box is displayed showing the individual records in that batch as shown in Image 4.

Image 4. Listing of Records Within a Batch

ID Fields	Stratum	Capture Source (9904)	Status	SEC	CritOvrFO	CritOvrHQ	CATI Rmks	Batch	LF
19 300080240 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300092840 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300093420 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300166660 1 1	1	NASS Keyed (2)	Suspect	0				543	
19 300183050 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300189260 1 1	1	CAPI (6)	Clean	0				543	
19 300318470 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300398590 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300406550 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300415960 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300436010 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300470140 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300602290 1 1	1	NASS Keyed (2)	Clean	0				543	
19 300679140 1 1	1	NASS Keyed (2)	Clean	0				543	

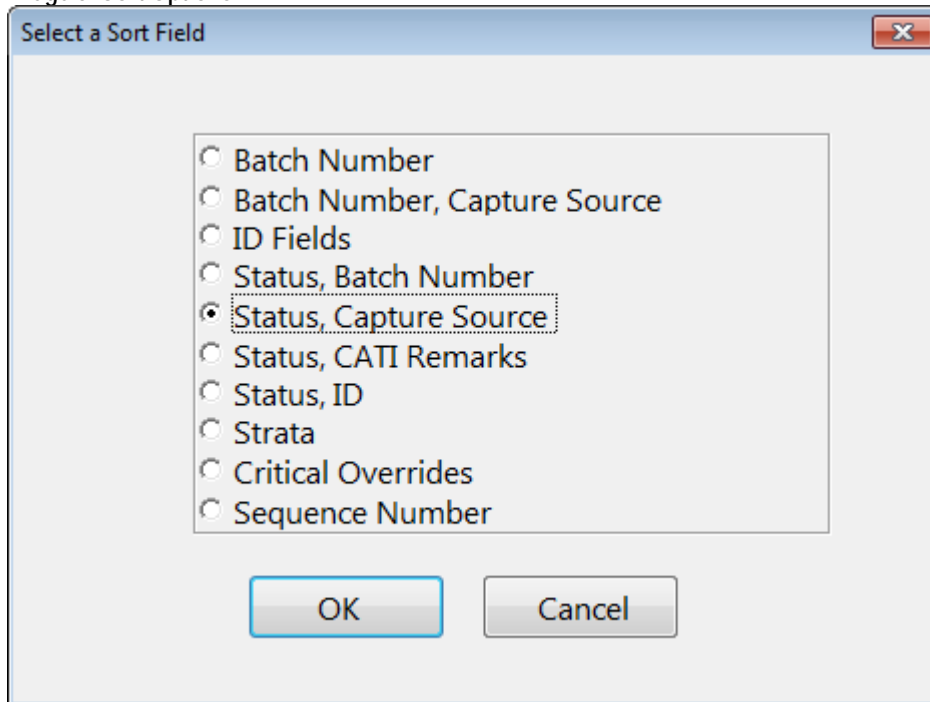
1:166

Edit Sort Cancel

Since records being read in with the automated program are assigned a batch relative to the date, there suddenly came the possibility of a batch being comprised of records that were collected by any combination of paper, EDR/CAWI, and CAPI. In the past when the users had control of assigning batch numbers, they always kept the different modes separated because the way in which editors edit

the records varies slightly between collection mode. We ended up adding a couple of new sort options that included Capture Source so that these records would sort together as shown in Image 5.

Image 5. Sort Options



Selecting the Status, Capture Source sort option groups the records by status and then capture source. This will bring the dirty records to the top of each group in the table to be edited as shown in Image 6.

Image 6. Sorted Listing of Records Within a Batch

ID Fields	Stratum	Capture Source (9904)	Status	SEC	CritOvrFO	CritOvrHQ	CATI Rmks	Batch	LF
19 800668700 1 1	1	NASS Keyed (2)	Dirty	0				543	
19 867282830 1 1	1	NASS Keyed (2)	Dirty	0				543	
19 877166700 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300019530 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300081260 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300404610 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300406180 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300416910 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300438450 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 300744500 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 772193140 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 772588320 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 772857020 1 1	1	NASS Keyed (2)	Dirty	0				543	
27 772918170 1 1	1	NASS Keyed (2)	Dirty	0				543	

1:166

Edit Sort Cancel

11. Improvement 5 - Email Alerts

When we first started using the new auto read in program, we kept a close eye on the log files being generated to be sure everything was running smoothly. At first this worked out well, as there were

numerous bugs to work out of the system. However as time went on and the bugs were getting fewer and fewer, no one wanted to keep tabs on the log files anymore. But as with any automated program, there must be some type of notification system in case things go awry.

Modifications were made to the auto read in program to automatically send emails to our group whenever any of the following conditions are detected:

- 1) An InUse.txt file is detected, so the assumption is that the Manipula program is still running and the data will wait until next sweep.
- 2) A Manipula failed to compile or some other problem was encountered when trying to complete the auto read in process.
- 3) No data files were found associated with the trigger file(s).
- 4) A file already exists to which the data files were to be concatenated.
- 5) An error occurred when trying to concatenate the files.

Once an email is received, steps are taken to resolve the problem and get the files set back up for the next sweep.

12. Challenges

Ever since we first deployed the auto read in process, we received random COMexception errors which would cause the read in to fail. Data files would remain in limbo in the TriggerStage and QueueStage folders, halting the read in process for that particular survey until someone in our group manually re-queued the data and deleted the InUse.TXT file. Putting the same exact data files back in queue would do the trick and we might not see another COMexception error for days or even weeks. There didn't seem to be any rhyme or reason to when they would occur. They'd show up across different surveys, different times and days of the week, and across the various servers we use. We were never able to recreate the error on demand, because as mentioned even queuing up the same exact files again wasn't able to recreate the error.

Documentation was created to instruct members of our group how to properly re-queue the data files. This worked; however, manually re-queuing the data files was a general annoyance for everyone to have to deal with. Also if someone didn't follow the re-queue instructions to the letter, it would cause further headaches with the process.

We eventually added extra code to the auto read in program to detect the random COMexception error and programmatically handle all of the re-queuing steps for us. Image 7 below shows an example of the log entries created after receiving this error at 7:00:02. It took a mere 1 second for the program to re-queue the data files, which were then successfully read in the next time the process kicked off 15 minutes later.

Image 7. Sort Options

```
12/29/2014 7:00:01 AM.... START of EDR ReadIn Process for BEEPDI140000.  
EDR data = 18737 bytes, Remarks = 689 bytes  
12/29/2014 7:00:02 AM.... ERROR in ReadIn Function while survey = BEEPDI140000.  
System.Runtime.InteropServices.COMException (0x8000FFFF): Catastrophic failure (Exception from HRESULT: 0x8000FFFF (E_UNEXPECTED))  
at StatNeth.Blaise.Interop.BIPars3A.IBlaiseParser.Parse()  
at AutoReadIn.ReadIn.ReadIn(String survey) in C:\VisualStudioProjects\CASICMenu\src\AutoReadIn\ReadIn.vb:line 834  
12/29/2014 7:00:03 AM.... COMException error occurred while survey = BEEPDI140000.  
Only EDR data was detected for requeue process (no RAW).  
12/29/2014 7:00:03 AM.... EDR data was successfully requeued for survey = BEEPDI140000.  
12/29/2014 7:15:01 AM.... START of EDR ReadIn Process for BEEPDI140000.  
EDR data = 18737 bytes, Remarks = 689 bytes  
12/29/2014 7:15:07 AM.... END of EDR ReadIn Process for BEEPDI140000.
```

We would love to get rid of this error altogether, but in the meantime at least we have a workaround in place that allows us to live with it without too much hassle.

13. Future Plans

Seeing how successful the auto read in became made us stop to think about whether there are other manually run processes we could consider automating. Some possibilities include our check in and CATI/Non-CATI transaction programs.

Something else we are investigating is incorporating an automatic pull of the EDR/CAPI data into the beginning of the overnight auto read in process. This would allow our users to see the most current data coming in from these two modes first thing in the morning without needing to manually pull the data and queue it up for the auto read in process.

14. Conclusion

The day we converted from the old method of reading data into Blaise to the new automated process marked a significant day in recent NASS data collection and editing history. General Blaise performance magically improved; service freeze occurrences all of a sudden dropped drastically in number; and user satisfaction with the system greatly increased because they suddenly had extra minutes, and in some cases hours freed up from their days without having to watch the data read process tick away on their screens or deal with other sluggish processes.

The new auto read in program also freed up some of our group members' time. Where we were previously spending significant amounts of time tracking and troubleshooting service freezes and overall poor system performance, we could now get our minds focused on developing other system improvements that previously had been put on hold. This system enhancement also gave us some ideas and coding experience to start thinking about other related improvements we could make in the future.

A Technical Solution for a Mixed-Mode Study

Hueichun Peng, Andrew Hupp, Jas Sokhal, Max Malhotra, University of Michigan

Abstract

The Pre-Post Deployment Study (PPDS) is a longitudinal component of the Army Study to Assess Risk and Resilience in Servicemembers (Army STARRS). Approximately 9,000 participants were included in the study. The most recent wave consisted of a mixed-mode design utilizing a self-administered web survey (Blaise IS) and an interviewer-administered phone survey (Blaise 4.8).

The study design led to the creation of a new sample management system. The new system was integrated with the existing Blaise CATI call scheduler used by the call center. The Blaise CATI call scheduler managed the cases assigned to phone, while the new management system managed cases assigned to web, mail, e-mail and SMS text communications, the safety protocol, the tracking protocol, and incentives. The new system was the repository for all study information, including information from the Blaise CATI call scheduler.

Topics covered in this paper include BlaiseIS management, e-mail/SMS text message management, mode switching processes, data consolidation between two systems, and tracking. It also includes lessons learned over the course of the study.

Background

The Pre-Post Deployment Study (PPDS) is a longitudinal study with a sample of approximately 9,000 participants. The first wave consisted of a computerized self-administered interview (BlaiseIS) and a blood draw, all in a group setting on the installation prior to deployment. The second wave consisted of a paper self-administered interview and a blood draw, all in a group setting on the installation immediately after returning from the deployment. The third wave consisted of a computerized self-administered interview (Blaise IS) in a group setting on the installation. The fourth wave asked participants to complete a self-administered web survey (BlaiseIS) or an interviewer-administered phone survey (Blaise 4.8).

Participants were assigned to be invited in one of two modes, web or phone. Cases with only a phone number were assigned to the phone mode. Cases with only an e-mail address were assigned to the web mode. Cases with both a phone number and an e-mail (the majority of cases) were randomly assigned between phone and web. The majority of cases were assigned to begin on the web. Those participants received an e-mail invitation and SMS text notification. An experimental design switched the mode on non-response cases to phone after 7, 14, 21, or 28 days. The majority were switched to phone after 21 days. Participants initially assigned to the web mode received e-mail reminders and SMS text notifications once a week until they were switched to telephone.

Non-final cases moved to tracking once all contact information was deemed to be invalid. To be invalid, an e-mail had to bounce back as undeliverable and/or the phone number was no longer valid. When new information was found the case went back to the production environment to be worked. A case could move in and out of tracking as new information was deemed invalid and new leads were discovered.

The phone and web instruments contained much of the same content but were not identical. If a case began in one mode and was accessed in the other mode, the data were pushed to the other mode for the participant to continue the survey. Questionnaire sections that were incomplete were restarted so questions in a particular section always had the same reference period. Once a survey was completed the other mode was no longer available.

The design included a safety plan. Answers to certain questions in the instruments could trigger a follow-up to make sure the participants are ok. An algorithm ran to identify cases that needed clinical follow-up. A participant could also request to speak with someone during the course of the phone interview.

A management system (PPDS-MS) was created to manage the overall workflow of sample between web and phone. When a case was assigned to phone, a legacy sample management system (Blaise SMS) programmed in Blaise was used to manage the calling activities. The PPDS-MS controlled the activation of sample; managed the cases assigned to web; consolidated data from Blaise SMS; and managed all e-mailing and SMS text message activities, tracking activities, mode switching from web to phone, safety plan activities, and respondent payments.

Timed release and switch-mode protocol

The sample was randomly assigned to replicates released over an eight month period. This allowed control of the flow of cases to the call center as web non-response cases switched to phone for follow-up.

Each release of replicates was assigned to one of the following groups:

- Phone only - cases providing partial contact information, lacking an e-mail address
- Web only - cases providing partial contact information, lacking a phone number
- Web first - cases providing full contact information (name, phone number, mailing address and email address)
- Phone first - a random sample of cases providing full contact information

Each release was governed by a complex set of experiments involving pre-notification and reminder activities and the timing of mode switching between web and phone.

Technical components

The PPDS-MS system included several modules, with the main database on an SQL server. Programming languages include ColdFusion for the primary Web System, C#.Net for Windows batch processing and modules, extensive use of T-SQL stored procedures, and a third-party interface for sending and receiving SMS texts (Aerialink). Below is a brief list of the key components of the PPDS-MS system.

- Web system for managing case attributes, dynamic reporting and query, tracking updates (Coldfusion)
- Windows batch processing program to deliver files to the Blaise SMS (C #.Net)
- Windows program to handle sending e-mails and logging undeliverables (bounce backs) (C#.Net)
- Windows program to handle sending text messages(C #.Net and Aerialink API)

- Windows program allowing staff to send customized text message replies to participants on demand (C #.Net)
- Extensive use of SQL Server procedures for batch processing

Controlled Web Mode

The PPDS-MS managed cases assigned to web. Respondents logged into the web survey via a web portal. The web portal validated the login credentials entered against the participant's login credentials in the PPDS-MS. If the credentials were validated and the case was active, the web portal redirected the case to the BlaiseIS web survey. The web survey interfaced with the PPDS-MS during the initial start of the survey and during exiting or completion of the survey so that the PPDS-MS had the most current status for that case. At the end of the web survey respondents were asked to provide contact information and explicitly asked if they were willing to accept a monetary token of appreciation for the time spent completing the survey. Upon the completion of the web survey; updated contact, payment, and safety information was pushed from Blaise IS to the PPDS-MS.

Communication to consolidate data between Blaise SMS (phone) and the PPDS-MS (web)

Since the PPDS-MS controlled operations of the web, and Blaise SMS controls operations for the phone mode; the technical challenge was how to dynamically communicate between the systems in as close to real time as possible. A case needed to be available in both modes at all times regardless of current mode assignment. Mode assignment was primarily controlled through the PPDS-MS. When a case was assigned to phone, there was focused effort to call the case to complete the interview over the phone. However, when a case was assigned to phone, it was still accessible on the web via the web portal. If a case completed the web survey, the updated status needed to be communicated from the PPDS-MS to Blaise SMS as quickly as possible to avoid a telephone interviewer attempting contact. Conversely, if a case was assigned to web and contacted the call center and completed the survey over the phone, this updated status needed to be communicated from Blaise SMS to the PPDS-MS as quickly as possible to avoid the system attempting contact (e-mail and SMS text).

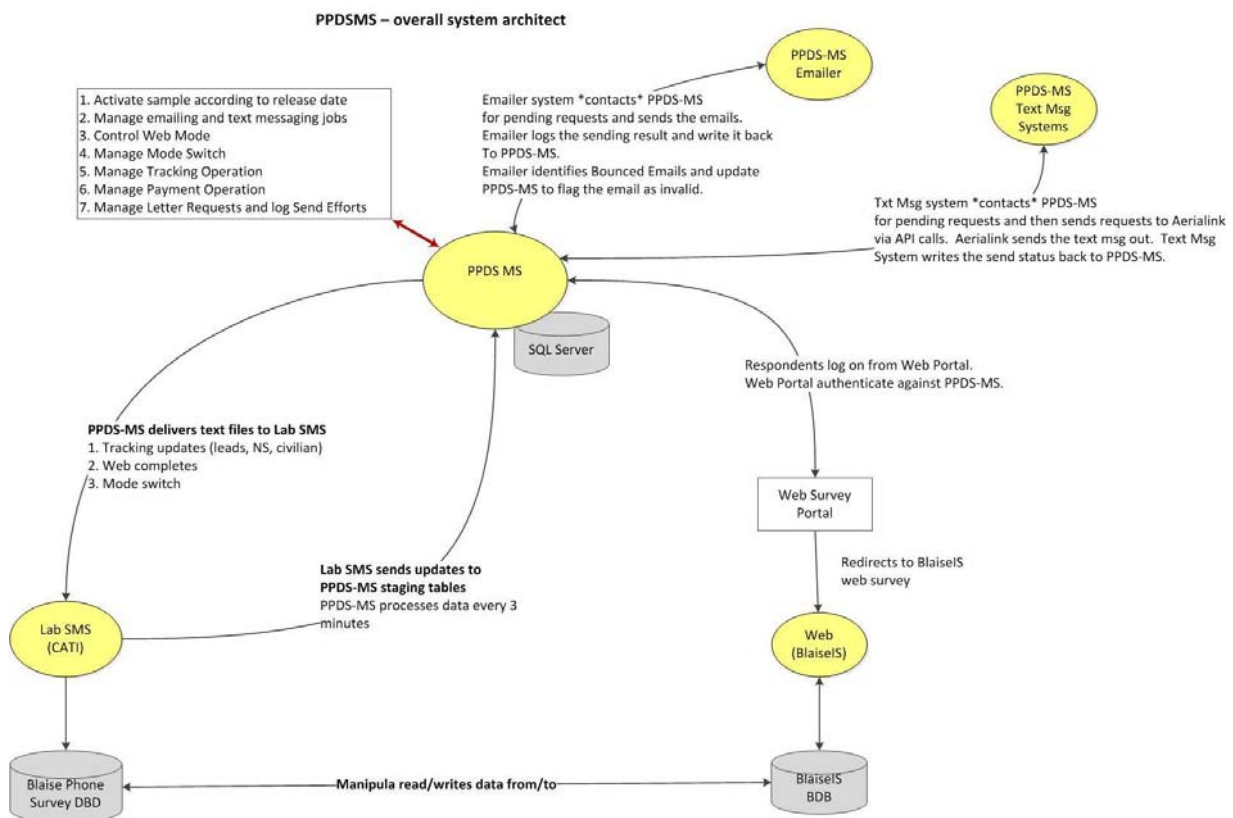
To facilitate communication, Blaise SMS pushed updated information to the PPDS-MS when a phone interviewer finished a case. The PPDS-MS had a scheduled job (SQL Server Stored Procedure Job) running every 3 minutes processing updates and taking appropriate action for a case. For example, during a phone interview, a respondent may have expressed a desire to receive an e-mail and provided a new e-mail address. Blaise SMS added a contact record signaling this action and sent the contact record along with the new e-mail address to the PPDS-

MS. After receiving this request and the new e-mail address, the PPDS-MS generated an on-demand email request. Requests were accumulated and triggered together with the automatic Batch Email job.

The PPDS-MS delivered a variety of text files at different time intervals to Blaise SMS communicating case status updates and information for next courses of action. Text files included information on:

- mode switching from web to phone
- finalization of cases (e.g. completion, refusal, lost, deployed during field period, etc.)
- mode re-activation and information updates from tracking

The diagram below describes the mixed mode management system architecture:



E-mail Job Management (sending, undeliverable (bounced) logging, throttle send speed)

The PPDS-MS managed the sending of e-mails (invitations and reminders) to cases assigned to web. Non-experimental cases were sent an e-mail invitation and three e-mail reminders. Experimental cases were sent an e-mail invitation and zero to three e-mail reminders depending on the experimental assignment. E-mail template types varied.

1. Different templates for invitation, reminder and final reminder (before switching to phone).
2. Templates varied based on e-mail domain. The “.mil” domain received a text e-mail, with a .pdf attachment with the same content containing the study and partner organization logos. All other domains (e.g., .com, .edu, .net., .org., etc.) were sent an HTML e-mail with the study logo and partner logos in the e-mail.
3. Templates varied based on experimental condition assignment. One group received an offer to call in to complete the interview via phone.

There were a total of twelve different e-mail templates. For each release, one of the twelve templates was assigned to a case. The PPDS-MS automated this process, eliminating manual intervention.

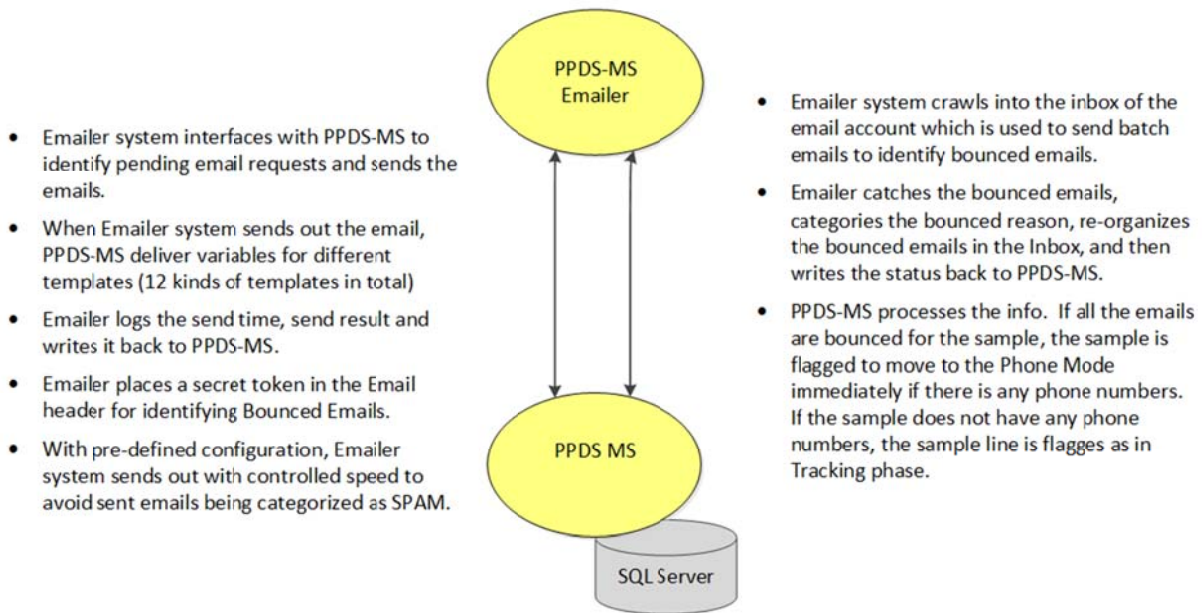
The PPDS-MS EMailer Utility was programmed in C#.Net and was created to send e-mails and log undeliverable (bounced) e-mails. Each morning, the PPDS-MS generated e-mail requests which were queued to be sent that day. The EMailer was scheduled to run each afternoon. It interfaced with the PPDS-MS to retrieve the generated e-mail requests and created a “sending job” with the .Net e-mail framework. When the EMailer sent an e-mail, it obtained the e-mail template type from the PPDS-MS to know which version of the e-mail to send. A token was added to the header of outgoing e-mails to identify and log an e-mail if it was undeliverable. The sending job had a throttle feature that was configurable. The throttle feature was configured with e-mails being placed in small batches with a 10 second pause between each batch and the rate (speed) between e-mails within a batch set at ½ second. This was done to minimize the chances of e-mails being marked as spam. When the sending job was triggered, it logged any error/failures encountered during the sending process. When sending failed, the error status was sent back to the PPDS-MS, and the e-mail was flagged as invalid. If there was no error, the status was posted back to the PPDS-MS with a timestamp.

Each night at midnight, the PPDS-MS EMailer triggered undeliverable (bounced) logging. The module scanned through the inbox of the sending e-mail address via the POP3 protocol. Using the token added to the header of the original (outgoing) e-mail sent from the EMailer, the EMailer logged undeliverable (bounced) e-mails and tracked them back to e-mail sent. The EMailer sent undeliverable (bounced) e-mail information back to the PPDS-MS and flagged the e-mail(s) as invalid.

With approximately 9,000 participants with multiple e-mail addresses, the automatic flagging of undeliverable (bounced) e-mail addresses saved significant operational time, eliminating the need for a manual review.

The diagram below illustrates the interaction between the PPDS-MS and the EMailer.

PPDSMS – EMailer Module



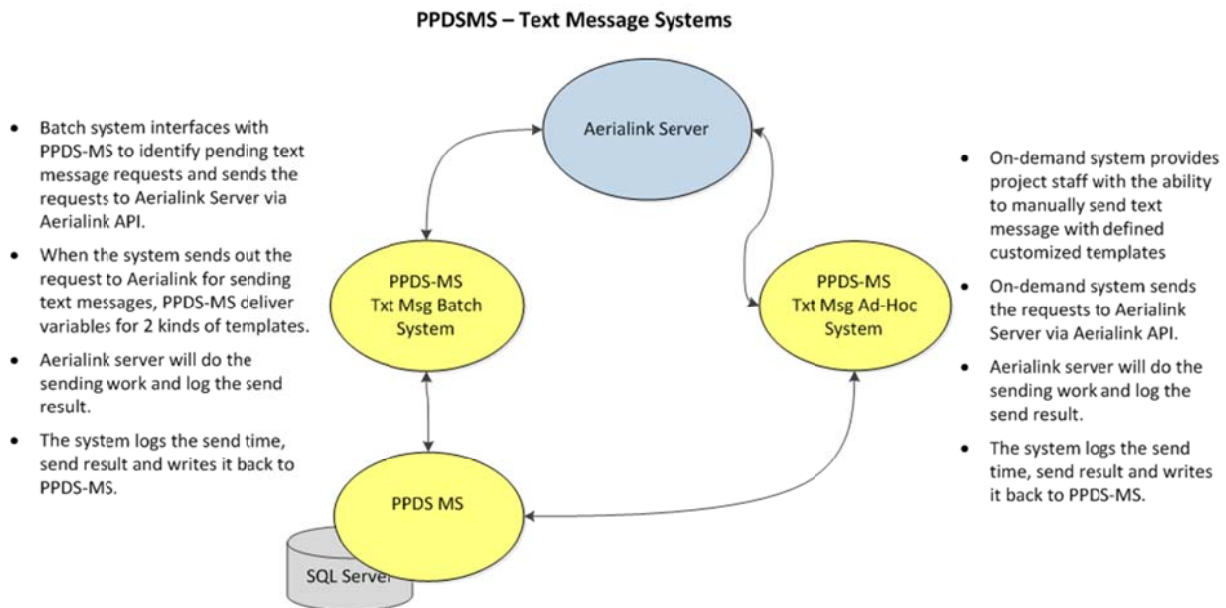
Text Message Management

The PPDS-MS also triggered the sending of SMS text messages to cases. Unlike e-mails, a SMS text message notification was sent to cases assigned to both phone and web when a phone number was available. The PPDS-MS used an external service provider (Aerialink) to send and receive SMS text messages. Aerialink provided a communication report of SMS text message failures/successes, and logged inbound text message replies from a case. Two sub-systems were created for this functionality: a Windows Batch System to trigger sending batches of SMS text messages, and an on-demand system (Windows Form application) for staff to reply to inbound SMS text replies from a case. Both sub-systems utilized the Aerialink API to send SMS text message requests to Aerialink servers. When the Aerialink API replied with an SMS text failure, the phone number was flagged as invalid in the PPDS-MS.

Similar to the way e-mail request lines were generated, the PPDS-MS generated a Batch Send job of SMS text message request cases each morning. The Batch System interfaced with the PPDS-MS to get the requested cases and information and made a call to the Aerialink server to send out SMS text messages. This processing was scheduled to run each afternoon following the e-mail sending batch job.

The Batch Send job and the on-demand manual send job both interfaced with the PPDS-MS to obtain current case status, recorded the send activity as a contact record, logged the send result and flagged out invalid phone numbers.

The diagram below illustrates the communication between the PPDS-MS and Aerialink.



Mode Switch

The PPDS-MS managed the complex cycle of mode switching. Web non-response cases switched to phone according to their experimental assignment. When it was time for switching to occur, the PPDS-MS Windows batch processing application delivered a switch-mode text file to Blaise SMS. The text file included the case ID and any updated contact information. When Blaise SMS processed the switch-mode text file each morning, it released cases into the calling pool for the phone interviewers. If there was any updated contact information, such as a new telephone number, mailing address or e-mail address; Blaise SMS updated its information from the text file.

Since web was always accessible to active cases, the PPDS-MS delivered a text file of cases completed on the web three times each day to Blaise SMS. Blaise SMS finalized those completed cases to remove them from the active calling pool.

For cases switching from phone to web, an interviewer entered a contact record in Blaise SMS. When the PPDS-MS identified the case to switch from phone to web, it activated and started the case with the correct communication (e-mail and SMS text message) protocol. If no valid e-mail address existed when the PPDS-MS received this notification, it triggered the case being assigned to the tracking protocol.

Respondent Tracking

When communication efforts, including calling and e-mailing were exhausted; a case moved to tracking and followed a five step protocol. About 30% of cases were tracked during the most recent wave of data collection.

For a case assigned to phone, an interviewer had to first confirm that all available phone numbers were invalid. Once all phone numbers were deemed invalid, the interviewer recorded a contact record that the case needed tracking. When the PPDS-MS received and processed this tracking contact record, it updated all phone numbers associated with a case as invalid. The PPDS-MS then checked to see if there was a valid e-mail address for the case and if it had previously been assigned to web. If it had not yet been assigned to web, the PPDS-MS switched the case to web. If no valid e-mails existed or the case had previously been to web, the case was assigned to tracking.

For a case assigned to web, the Emailer Bounce Back logging triggered each night. Any e-mail addresses that were undeliverable (bounced), the PPDS-MS updated to invalid. At this time, another processing procedure was triggered.

- If the case had other valid e-mail addresses, the case remained assigned to web.
- If the case had no other valid e-mail addresses, and had valid phone numbers, it switched to the phone immediately.
- If the case has no other valid e-mail address or valid phone numbers, it went to tracking.

Once cases were flagged for tracking, the PPDS-MS managed cases through the tracking protocol. There were a total of five steps in the protocol a case could go through. When a tracking lead was identified the case moved out of the tracking protocol and moved back to production. An updated e-mail address reactivated a case on the web, and an updated telephone number reactivated a case in phone, unless there was also an updated e-mail address which reactivated the case to web first. When a case reactivated in phone mode, a text file was delivered to Blaise SMS that activated the line and imported updated contact information.

Lessons Learned

1. The interviewer needs access to all of the contact attempts. The e-mail and SMS text message contact records were only stored in the PPDS-MS. The interviewers using Blaise SMS never had access to these records. Having the additional context is helpful to the interviewer so they can see how a case was previously contacted and can reference that they may have received an e-mail or SMS text message when contact is made. An

interviewer can also confirm an e-mail was sent if a case requested an e-mail in a previous contact.

2. Inbound SMS text messages need to be logged/recorded by the system. The PPDS-MS recorded outbound SMS text messages but not inbound SMS text messages (reply from a case). Staff monitored a separate e-mail account to check for inbound SMS text messages and manually logged those messages.
3. Having one management system managing all aspects would be more efficient than having two separate systems: one to manage the majority of the processes including web and a second system just to manage telephone calling. Data had to be pushed back and forth between two systems in a timely manner to ensure each system had the most up-to-date information.

Acknowledgements:

This research was supported by NIH Cooperative Agreement U01MH87981, with funds provided by the Department of the Army (\$50M) & National Institute of Mental Health (NIMH: \$15M). Army STARRS is being conducted by a consortium of investigators—from the Uniformed Services University of the Health Sciences (USUHS), the University of California-San Diego (UCSD), Harvard Medical School (HMS), & the University of Michigan (UM)—in collaboration with the NIMH. Co-PIs: R. Ursano (USUHS) & M. Stein (UCSD); Site-PIs: R. Kessler (HMS) & S. Heeringa (UM); Collaborating NIMH Scientists: L. Colpe & M. Schoenbaum; and Consulting Army Scientists: K. Cox & S. Cersovsky

The Michigan Questionnaire Documentation System (MQDS) Version 5 Update

Karl Dinkelmann, Peter Sparks, Jason Ostergren, & Gina-Qian Cheung, University of Michigan

1. Introduction

The Michigan Questionnaire Documentation System (MQDS) is an application that extracts data and metadata from Blaise instruments and generates various types of output. These outputs include codebooks, questionnaire documentation, and human readable metadata documentation. Additionally, MQDS can serialize the extracted information as a Data Documentation Initiatives (DDI) XML instance. DDI is an internationally recognized XML standard for documenting and disseminating social survey data and metadata. The introduction of the next generation of Blaise, version 5 and the recently released DDI version 3.2 standards offer a unique opportunity to rewrite MQDS to adapt to these changing technologies. MQDS version 5 is being rewritten to work with Blaise 5 data models and datasets using the Blaise 5 API (Application Programming Interface). MQDS 5 will also incorporate work done by Ostergren & Ash, 2013 allowing one to visualize the Blaise logic and dynamically manipulate it within a visual GUI interface.

We are taking a modular approach to constructing the components necessary for the next version. With MQDS 5, one goal will be to clean and unify the code. However, the core features of extracting Blaise metadata, generating statistics and frequencies from data, representing human readable output, and generating a DDI instance remain intact.

2. Background

For over a decade, Survey Research Operations (SRO), a unit within the University of Michigan's Institute for Social Research's Survey Research Center, has developed and enhanced MQDS to allow users to export Blaise metadata and data to document and disseminate Blaise questionnaires to their users. The development of the first version began in 2003 (Sparks and Liu 2004, Guyer and Cheung 2007). Sparks and Liu (2004) describe early development of MQDS version 1 and what was referred to as "BlaiseDoc" in the early stages of developments. Guyer and Cheung (2007) further explained development activities that lead to MQDS versions 2 and 2.5, rewriting the program in .NET, and described utilities. The creation of version 3 began in 2009 and was outlined in Dinkelmann et al (2009). Version 3 was database driven and written to use the DDI 3.0 standard. In 2011, MQDS 4 was enhanced to more efficiently process larger data models and the DDI 3.1 standard. All versions of MQDS 4 and earlier used either Blaise version 4.6, 4.7, or 4.8.

2.1 MQDS 3 & 4

The introduction of MQDS 3 added the feature of writing the Blaise metadata and data to a database. In theory, this would have been the ideal design. However, it was tightly coupled to the DDI 3.0 specification. Therefore, as new specifications were released, an in depth understanding of the changes was required in addition to the knowledge required to make the DDI 3.0 database schema work with the updated 3.1 and 3.2 DDI specifications. This led to limitations in the database design used in MQDS 3 and 4. In creating MQDS 5, we will no longer continue active development of MQDS 4. However, support will remain for MQDS 4 as needed moving forward. Depending on the adoption rate of Blaise 5, it is possible that MQDS 5 could support Blaise 4 in the future.

2.2 DDI 3.2 Standard

The Data Documentation Initiatives (DDI) XML version 3.2 adds features that will assist in creating DDI compliant metadata with MQDS. One of the primary features added to DDI 3.2 is the introduction of the *FragmentInstance* that allows one to generate DDI instances that are valid and only contain the actual

elements that are needed. In previous versions of DDI, required elements that were not in the Blaise metadata either needed to be captured or empty XML tags were written to ensure the XML instance validated to the DDI Schemas. Other features introduced in DDI 3.2 include, but are not limited to: modeling of the data flow throughout an instrument with input and output parameters as well as bindings, question grids, and scale questions. DDI Version 4 is already in the development phase although a release date has not been set.

2.3 Blaise 5

The next generation of the Blaise software created by Statistics Netherlands is Blaise 5. It brings forth many major changes in what survey developers are able to accomplish. It also has a completely new implementation of the Application Programming Interface (API). This is important, as all tools that currently exist around Blaise 4.8 will need to be updated to take into account the new structure of the API. Therefore, in order to access the metadata within Blaise 5 MQDS updates are needed using the new API.

3. MQDS 5

The guiding principles in creating MQDS 5 are as follows:

- Continued support in Blaise 5 for derived metadata information like fill resolution and gotos
- Standalone executables allowing for faster testing and easier reuse
- Internal use of XML tailored to Blaise vs. attempting to do processing within a DDI framework; DDI will be an output product instead
- Ability to support multiple DDI versions

As Blaise 4 matured, many new methods were exposed to allow one to generate Blaise metadata. In fact, Blaise Manipula became powerful enough to export metadata. de Bolster (2013) demonstrated how to export DDI instances using Manipulus. However, with Blaise 5 the data manipulation, re-coding, exporting and importing features of Manipula remain under construction. This also means that the tools within Blaise 5 for extracting data and metadata are minimal to date. One can export the meta information from the control center to an xml file within Blaise 5. However, to use the exported information one would need to create an XSL transformation stylesheet to create useful documentation or to any other XML specification representation (such as DDI). We have not explored using this file but it would be interesting to see how one could potentially use this file. Below is an example of exporting the “Flight Survey” from the *Meta View* (figure 1) and the resulting XML file that is created (figure 2).

Figure 1: Generating the Meta View XML file from Blaise 5

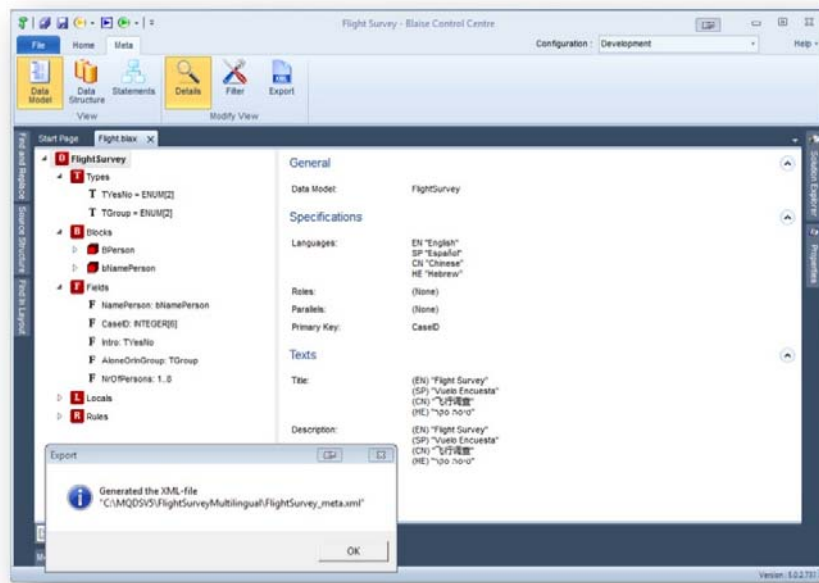


Figure 2: Example of the “Fight Survey” Meta XML File

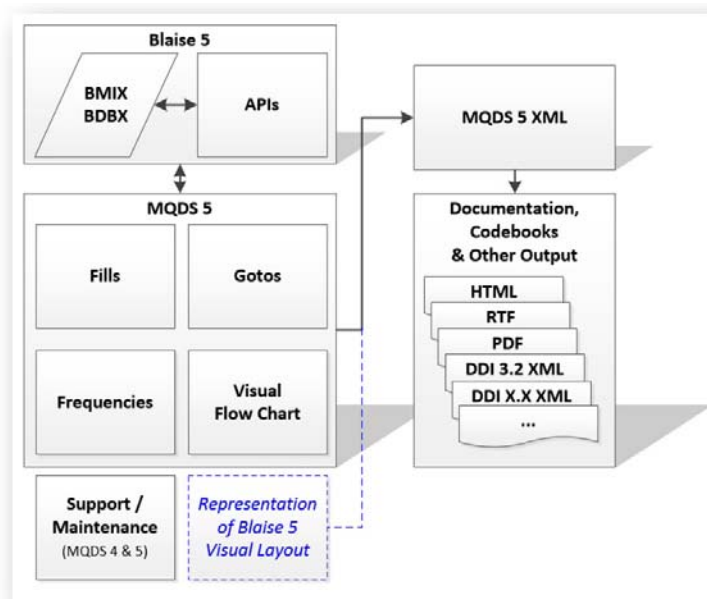
```

<!--Generated by Blaise 5.0.2.731 at 3/11/2015 11:15:04 AM-->
<DataModel Name="FlightSurvey">
  <RoleTextsCollection>
    <RoleTexts Role="Title">
      <Text>
        <Expression ExpressionStructure="Constant" TypeStructure="String" Text="Flight Survey" />
      </Text>
      <Text>
        <Expression ExpressionStructure="Constant" TypeStructure="String" Text="Vuelo Encuesta" />
      </Text>
      <Text>
        <Expression ExpressionStructure="Constant" TypeStructure="String" Text="飛行調査" />
      </Text>
      <Text>
        <Expression ExpressionStructure="Constant" TypeStructure="String" Text="המסע הראשון" />
      </Text>
    </RoleTexts>
    <RoleTexts Role="Description">
    </RoleTexts>
  </RoleTextsCollection>
  <LanguageCollection>
    <Language Name="EN" Description="English" IsActive="true" />
    <Language Name="ES" Description="Español" IsActive="true" />
    <Language Name="CN" Description="Chinese" IsActive="true" />
    <Language Name="HE" Description="Hebrew" IsActive="true" />
  </LanguageCollection>
  <KeyDefinitionCollection>
    <KeyDefinition Name="PRIMARY" KeyStructure="Primary">
      <NameExpressionCollection>
        <Expression ExpressionStructure="Name" Name="CaseID" />
      </NameExpressionCollection>
    </KeyDefinition>
  </KeyDefinitionCollection>
  <MetaDefinitionCollection>
    <MetaDefinition Name="TypeHo" DefinitionStructure="Type">
    </MetaDefinition>
    <MetaDefinition Name="TGroup" DefinitionStructure="Type">
    </MetaDefinition>
    <MetaDefinition Name="BPerson" DefinitionStructure="Block">
    </MetaDefinition>
    <MetaDefinition Name="BNamePerson" DefinitionStructure="Block">
    </MetaDefinition>
    <MetaDefinition Name="NamePerson" DefinitionStructure="Field" FieldKind="DataField">
    </MetaDefinition>
    <MetaDefinition Name="CaseID" DefinitionStructure="Field" FieldKind="DataField">
    </MetaDefinition>
    <MetaDefinition Name="Intro" DefinitionStructure="Field" FieldKind="DataField">
    </MetaDefinition>
    <MetaDefinition Name="AloneOrInGroup" DefinitionStructure="Field" FieldKind="DataField">
    </MetaDefinition>
    <MetaDefinition Name="HrOfPersons" DefinitionStructure="Field" FieldKind="DataField">
    </MetaDefinition>
    <MetaDefinition Name="Index" DefinitionStructure="Local">
      <ClaimsType TypeStructure="Number" Decimals="0" />
    </MetaDefinition>
  </MetaDefinitionCollection>
</DataModel>

```

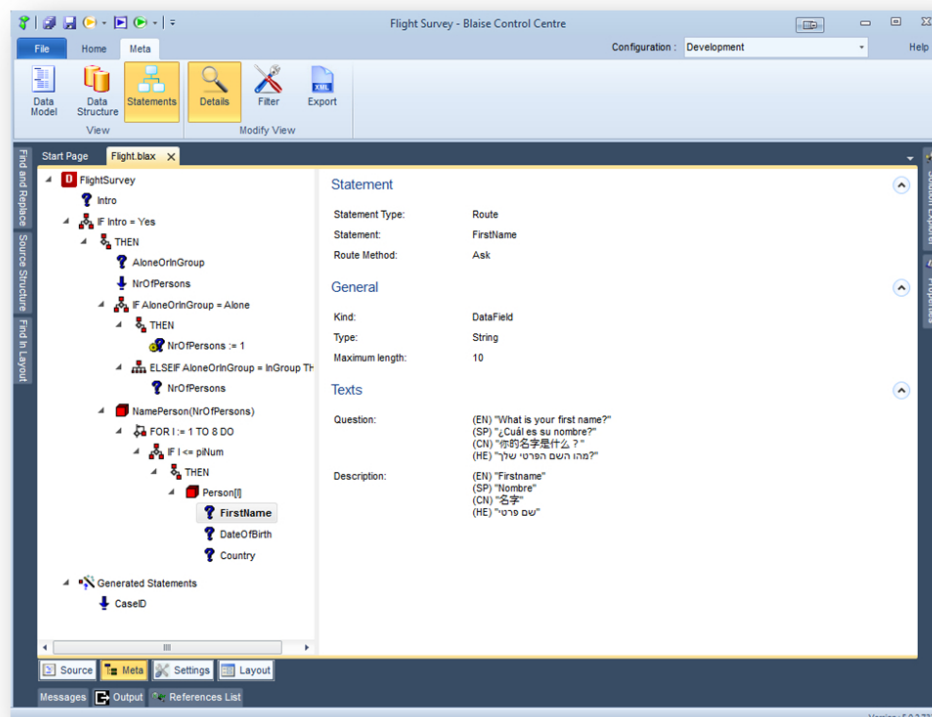
Using the DDI 3.X XML specification as the basis for the XML metadata documentation is not desired for similar reasons. The MQDS 5 XML is structured for fast retrieval of meta information. For these reasons, MQDS 5 will use its own XML schema. This also aids in the extraction of metadata and data in a format that assist in completing some of the main features of MQDS. Two of these features are its ability to resolve question fills and generate goto statement representation. These two features will comprise the core features included within MQDS 5. An initial hurdle in creating MQDS 5 was that we had few Blaise 5 instruments to use for testing. While as the number of production Blaise 5 instruments available for testing is still limited, a few large Blaise instruments exist that SRC is currently in the process of converting to use with Blaise 5. Additional components and features currently part of MQDS 5 can be seen below in Figure 3.

Figure 3: MQDS Version 5



Representing the visual layout of a Blaise 5 instrument is a feature that would be nice to support in the future. However, this feature will not be incorporated for some time due to reasons discussed in section 4. One of the newest and most exciting features is the inclusion of the Visual Flow Chart program developed by Ostergren & Ash, 2013. First demonstrated at the Washington, D.C. Blaise Users Conference, this feature has been updated to use Blaise 5. Blaise 5 itself allows users to drill into an instrument within the *Meta View* inside the control center. However, the *Meta View* offers a static treeview-style visual representation of the data model. Figure 4 shows how this looks in the *Meta View* within Blaise 5 when selecting the *Statement* view.

Figure 4: Meta Statement View in Blaise 5

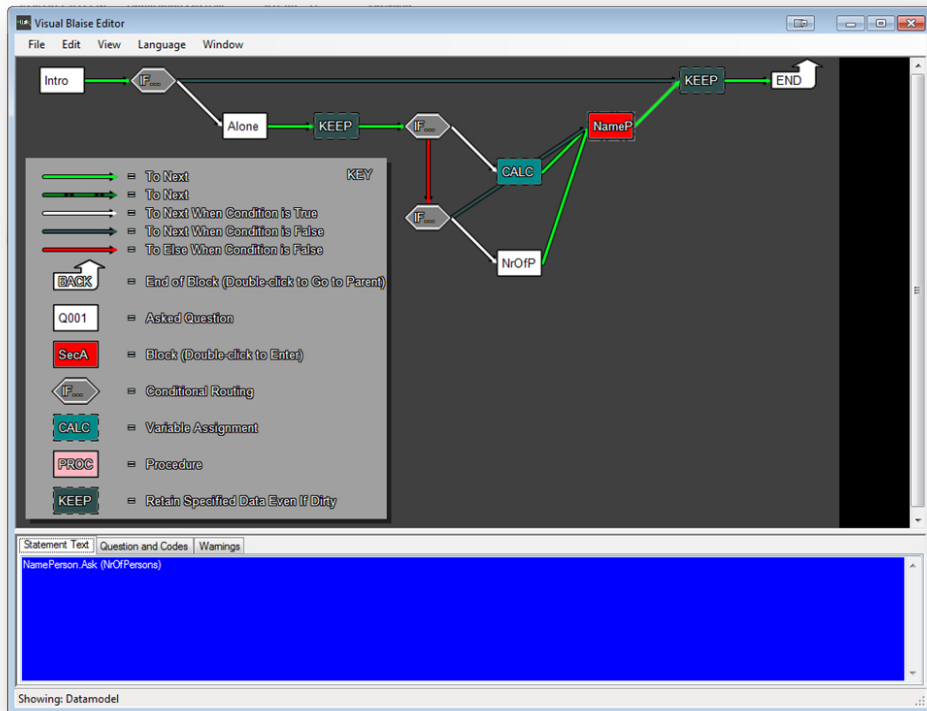


Ostergren and Ash (2013) describe their Visual Flow Chart program (known internally as “Visual Blaise”) as :

“... an application that graphs the logic of a Blaise instrument one block at a time. It allows elements to be dragged and dropped, cut and pasted, and allows for the addition of new elements. Finally, it is capable of exporting a text file with the logic for each block written as a RULES section in correct Blaise syntax.”

A graphical representation of the same “Flight Survey” code shown in figure 4 is displayed in figure 5.

Figure 5: Visual Flow Cart in MQDS 5



The main purpose of this flow chart is to provide an alternative visual representation which is easier to understand in many cases. One other difference between *Meta View* inside Blaise 5 and the Visual Flow Chart program is that the user can drill in and out of blocks and dynamically change the block structure, questions placement, and instrument logic. These changes can be exported to a text file which can be used to update the rules in the code.

4. Discussion of Potential Issues & Future Exploration

Perhaps one of the core differences between Blaise 5 and Blaise 4 with regards to the metadata is that it divides the survey instruments' core metadata from the layout. Blaise 5 does this by placing the layout information in a separate .layout file that is used to map between the metadata and the Resource Database. This adds to the complexity of documenting the look and feel of a given survey instrument. Blaise 5 features have far-reaching capabilities allowing survey developers to modify virtually all the aspects of the presentation of surveys for the end user. With all previous versions of MQDS the default presentation of the output was to mirror the Blaise 4.8 split screen layout presentation. However, the introduction of Blaise features allowing one to create their own visual presentation from the ground up will present some obstacles for one to programmatically document the actual presentation as seen by the end user. Layers of complexity are added with the addition of text roles, grouping, parameters, applicability conditions, and many other layout concerns that can all be impacted differently depending on the target device, platform, and mode. Therefore, accurately and adequately documenting Blaise 5's end users visual presentation will not be simple. To do this programmatically, further exploration and expansion of the API may be needed.

To what extent do these features need to be documented and/or disseminated? This is not something that is necessarily needed by all end users. However, this may be something that those who research effects of the visual presentation of instrumentation will want and require that they be documented.

It is possible that MQDS 5 would support Blaise 4 data models, although this is not currently on the development schedule. One of the main advantages of MQDS using its own XML standard is that it allows one to create any DDI standard. This makes it possible to only add additional modules instead of needing to rewrite the entire program when wanting to export to multiple versions of the DDI standard, regardless if it is a newly supported version or an older version.

5. Summary

MQDS 5 will continue to support the extraction of Blaise 5 data and metadata and will support new methods of viewing and modifying the Blaise 5 logic through the inclusion of the visual layout editor. As Blaise 5 evolves additional features will be identified and considered. The initial phases of development have begun on MQDS 5 and a pre-beta version release is expected by April 2015, with the first production release expected in summer 2015. Questions regarding MQDS 5 can be submitted to SRC.MQDS-5@umich.edu.

6. References

- Blaise 4.8 Online Assistant (Version 4.8.4.1899). (2015).
- Blaise 5 Help (Version 5.0.2.731). (2015).
- DDI Lifecycle 3.2 <http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/>, Retrieved March 10, 2015.
- de Bolster, G. (2013). Generating Blaise from DDI. *The 15th International Blaise Users Conference (IBUC)*, Washington, DC.
- Dinkelmann, K. (2006). DDI Version 3 and Instrument Documentation. *The proceedings of the 32nd International Association for Social Science Information Services & Technology (IASSIST)*. Ann Arbor, Michigan.
- Dinkelmann, K., Kirgis, N., and Cheung, G. (2009). Michigan Questionnaire Documentation System, Version 3 (MQDS-V3). *The proceedings of the 12th International Blaise Users Conference (IBUC)*. Riga, Latvia.
- Guyer, H. & Cheung, G. (2007). Michigan Questionnaire Documentation System (MQDS): A User's Perspective. *The proceedings of the 11th International Blaise Users Conference (IBUC)*. Annapolis, Maryland.
- Ostergren, J. & Ash, R. (2013). A New Tool for Visualizing Blaise Logic. *The 15th International Blaise Users Conference (IBUC)*, Washington, DC.
- Sparks, P. & Liu, Y. (2004). Blaise Documentation System. *The proceedings of the 9th International Blaise Users Conference (IBUC)*. Gatineau, Québec.
- Thomas, W., Gregory, A., Gager, J., Johnson, J., and Wackerow, A. (2015) Data Documentation Initiative (DDI): Technical Specification Part 1: Technical Document Version 3.2. © DDI Alliance 2015.

Research And Development of EHC:The .NET Questionnaire System in EHC

Shuai Sun, YongJian Zhang, JiaHui Yao, iSSS of PKU

1. Introduction

China Family Panel Studies (CFPS) aims to reflect the transition of Chinese society, economy, population, education and health by tracking and collecting data of three levels-individuals, families, communities, providing the data base for academic research and public policy analysis.

The Event History Calendar(EHC) is kindly used as another effective way to solve the complex information collection. Interviewee need to recall the related information for the questions, and the longer the recall period, the lower the accuracy of the information. EHC is an effective tool to help interviewee to better recall time information. Because the frequency of follow-up interview changed from originally once one year to two years, since 2014, CFPS has been used the EHC collecting information in three modules—the change of residence, work, marriage, in order to improve the accuracy of the information.

EHC was developed by the mainstream object-oriented programming language, applying to adult questionnaire of CFPS(China Family Panel Studies),to help interviewee to recall questionnaire answers and interviewers to input information conveniently through visualization and time table. In addition, it could also checkout the reasonability of the data by software check, hardware check, and overall logic checks for increasing the efficiency of data collection.

CFPS EHC has its own independent questionnaire system, which used dynamic configuration form in its question and jump logic, and its data will be written in the Blaise bdb synchronously to ensure the consistency of Blaise data and EHC questionnaire data.

2. Background

CFPS PI team began planning to add EHC module into the questionnaire of CFPS in 2012, and determine its needs in 2013. iSSS execution team then begin the work of EHC design and development, during the process, we learned the EHC program of PSID, read some articles about EHC finding from Blaise forum, which gave us a lot of inspiration but also let us take a lot of detours less.

After nearly a year of design and development work, we first used the EHC module during the execution of survey in 2014, EHC program runs relatively stable,and we carried out large-scale version of the update only once during the survey in 2014, but it's just one aspect of the evaluation for EHC program,it needs the assessment of data analysis team and PI team for EHC program

Figure 2: EHC questionnaire module

```

<Question>
  <QID>AR1</QID>
  <QName>AR1</QName>
  <BQID>EAR1</BQID>
  <BQFullName>EHCR.EAR1</BQFullName>
  <BQIfArr>False</BQIfArr>
  <QType>SingleSelection</QType>
  <QDK>False</QDK>
  <QRF>False</QRF>
  <QShowCard>False</QShowCard>
  <QHelp>False</QHelp>
  <QShorStem>2012地址确认</QShorStem>
  <QStem>根据我们的记录，您_CFPS2012_time调查时是在_residence2012居住，是吗？</QStem>
  <DicQOptions>
    <Option name="1" value="是"/>
    <Option name="5" value="否"/>
  </DicQOptions>
  <QAttention value="" />
</Question>

```

Figure 3: XML questionnaire

```

4    <Question name="AR1" jump="AR101">
5      <Answer name="1" jump="AR102"/>
6      <Answer name="5" jump="AR101"/>
7    </Question>

```

Figure 4: XML logic jump

4. Unresolved Issues

There may be data redundancy problems in the EHC questionnaire system, all the questions interviewee answered will be stored in the current version of the EHC program, but there may be some questions ineffective. The causes of the problem are as follows:

1. EHC questionnaire system is too complex , different in the questionnaire system preload data and different questions answers can cause different question Jump.
2. Interviewer can return to modify the answer, this will lead to questions already filled out becomes invalid.
3. Interviewers can jump to a question by double-click a topic in the navigation bar on the left of the EHC questionnaire system. (The navigation bar will show the navigation information of question completed only)
4. EHC system did not consider saving mechanism for data storage of invalid questions, when save data,all of the question answers will be saved.

Considering that the issue is complicated to modify, and worry about the safety of data ,the problem is not resolved in the current version, we will consider in the next version to solve this problem.

5. Data output and format

There are three EHC data format, the first type is Blaise data(BDB), all of the EHC questions will be stored in the questionnaire BDB, so to ensure the consistency of EHC data and Blaise questionnaire data.

The second type is the SQL Server data, this section of data applies to EHC migration module. In relation to five addresses (provincial, city, county, township, Murai) selection, EHC directly read the SQL Server data of IM system, This ensures consistency of data interviewer see.

The third type is XML data, The trace data which EHC recorded will be stored as XML files. An XML file will be stored locally in the EHC at the completion of each sub-block problem, The file records the interviewer trace data during operation EHC, but the current record of the trace data is limited to the answers of the question and the submit time of the question, No Blaise ADT documented less detailed.

```

<EHCDData>
  <DataSet ModuleName="JobData" EnterTime="9-07-26" SaveTime="9-08-08">
    <JobData>
      <GC1051 BQID="EGC1051" QType="UserInput">在家务农种地</GC1051>
      <MAINJOB_2014 BQID="MAINJOB_2014" QType="UserInput">在家务农种地</MAINJOB_2014>
      <GC1052 BQID="EGC1052" QType="StartTime">
        <QRStartYear>1900</QRStartYear>
        <QRStartMonth>1</QRStartMonth>
      </GC1052>
      <GC1053 BQID="EGC1053" QType="EndTime">
        <QREndCurrent>True</QREndCurrent>
        <QREndYear />
        <QREndMonth />
      </GC1053>
    </JobData>
  </DataSet>
  <DataSet ModuleName="JobData" EnterTime="9-07-26" SaveTime="9-08-15">
    <JobData>
      <GC201 BQID="EGC201" QType="UserInput">0</GC201>
    </JobData>
  </DataSet>
</EHCDData>

```

Figure 5: Trace data format

6. User-friendliness

6.1 Special symbols of the beginning month and the end month

The special symbols will appear at the beginning and the end of each period in the EHC calendar. "#" means the year and the month the interviewer selected; "*" represents the uncertain year the interviewer chosen; "?" represents the uncertain month the interviewer chosen. Figure 1: EHC's home page.

6.2 EHC marriage module distinguishes the marriage from the cohabitation by color

In the EHC marriage module, there will be questions about pre-marital cohabitation and marriage. The yellow color represents the period of cohabitation, if pre-marital cohabitation emerged before marriage, and the green color means the duration of marriage.

Similarly, in a separate marriage question block, the time period for the green color, in a separate cohabitation question module, the time period for the color yellow. Figure 1: EHC main interface.

6.3 Five address selection

Five- address selection module has been developed in the EHC migration module for recording the interviewee's migration address supposed to be the same with IM system, which shares the address data with IM system, in order to ensure the consistency of the data and improve the accuracy of address data entry.

Administrative regions in China is divided into five level: provinces, cities, counties, towns

(street), Murai (community), One of the superior administrative unit may contain multiple lower-level administrative units, and each lower administrative unit belongs to only one superior administrative unit, such as a province may have more than one city, but a city can only belong to a province. we have developed five addresses linkage module against this background and uses Standards data from the National Bureau, Interviewer just need to choose to start from the provinces. (Select the primary address, the next level will automatically filter, leaving only the lower address data which is Included in the selected address)Figure 6: The address five drop-down box.

Figure 6: The address five drop-down box

6.4 Death list selection

In marriage module of EHC, if the situation interviewee spouse dies, we will ask the cause of death, where in order to maintain consistency with Blaise questionnaire, considering the data's accuracy and the convenience for the Interviewer, the death list selection module has been added, interviewers in the topic can simply double-click the left mouse button or click the keyboard spacebar to bring up the death lists. Figure 7: A list of death EHC questionnaire module.

2012 婚姻

婚姻结束日期
分开原因

回答 帮助

对方是因为什么原因去世的?

死亡原因:

1.0.10

上一题 下一题

死亡原因列表

- 1 A. 传染病和寄生虫病小计
 - 1.0 传染病计
 - 1.0.1 伤寒和副伤寒
 - 1.0.10 败血症
 - 1.0.11 流行性乙型脑炎
 - 1.0.12 流行性出血热
 - 1.0.13 麻疹
 - 1.0.14 病毒性肝炎
 - 1.0.15 艾滋病
 - 1.0.2 痢疾
 - 1.0.3 肠道其他细菌性传染病
 - 1.0.4 呼吸道结核
 - 1.0.5 其他结核
 - 1.0.6 钩端螺旋体病
 - 1.0.7 破伤风
 - 1.0.8 百日咳
 - 1.0.9 脑膜炎球菌感染
 - 1.1 寄生虫病计
 - 2 B. 肿瘤小计
 - 3 C. 血液、造血器官及免疫疾病小计
 - 4 D. 内分泌、营养和代谢疾病小计
 - 5 E. 精神障碍小计

Code: 1.0.1

确定 取消

Figure 7: A list of death EHC questionnaire module

6.5 Message box

If interviewers put the mouse on any finished time region for two seconds, the system will automatically pop-up message box which displays the section title and the start and end time for the convenience of interviewers and avoiding the error due to the time period is too long or too far away from the first line of the date title.

手机贴膜

#

手机贴膜

2012年8月
到
2013年8月

Figure 8: message box

7. Summary

After more than one year's design and development, CFPS EHC program has experienced many difficulties with expected and unexpected questions, which has already been solved one by one through the joint efforts of the iSSS'S technical group and the help of the survey team in Michigan

and Blaise technical support team. We, during that process, not only learned the new technology, but more importantly, got a lot of problem-solving approaches.

Particular thanks to the Survey Center of Michigan University, for their selfless help on technology. Thanks to other organizations and individuals who developed and used EHC before us and you are our successful guides.

References

1. Development and Programming of an Employment Event History Calendar in the Panel Study of Income Dynamics

April Beale, Mary Dascola and Youhong Liu, University of Michigan

2. Customizable .NET Event History Calendar: Looking to the Future

Roberto V Picha, Daniel Moshinsky, Mecene Desormice, Seth Benson-Flannery, U.S. Census Bureau

3. The Use of Metadata in the Design of a Customizable .NET Event History Calendar

Daniel Moshinsky, Mecene Desormice, Seth Benson-Flannery; US Census Bureau, USA

Using the BCP for Creating SAS Datasets and Codebooks

Lilia Filippenko, Mai Nguyen, and Chris Carson, RTI International

1. Introduction

Blaise Component Pack (BCP) allows us to create applications that read meta information from the Blaise meta file. This paper describes a .Net application (BlzToSAS) developed at RTI International which uses BCP to prepare SAS datasets easily and efficiently for any Blaise instrument regardless of its complexity and size.

The application uses an initialization file and creates a few intermediate files along with a driver program for creation of the SAS dataset. These files can be used during the development phase of the instrument to examine test data and during the production phase to prepare deliverable SAS datasets. The initialization file is created dynamically and holds a number of settings such as maximum length of characters allowed in variable names and variable labels, requirements for renaming Blaise variables, and many others. The application is used to check how variable names from the Blaise database will be presented in the SAS dataset and helps with preparation of codebooks.

2. Background

The initial version of this application was developed in Microsoft Visual Basic 6.0 when one of our studies used a Blaise instrument with almost 35 thousand fields. RTI's customized Cameleon script was already used successfully for many studies but didn't work properly with that instrument. We decided to use Blaise API to create a simple application that would be easy to use and easy to adapt later for other studies. The application was created in a very short amount of time and worked very well to produce SAS programs to create datasets. Some ideas are still used in the application and many new features have been added to it during the last nine years. A few years ago it was converted to VB.Net and uses the latest version of the Blaise API. The application is used for all studies at RTI International conducting CAPI interviews in Blaise.

3. Instrument Development

3.1 Blaise Instrument Requirements

Some standards are established at RTI for developing Blaise instruments to simplify the process of creating SAS datasets by using the BlzToSAS application:

- **Field Description** is used to specify a label for a field in the SAS dataset or to define a field as not needed in the delivery file. The latter is achieved by using key word "NoDeliv".
- **Field Tag** is used to define a variable name in the SAS dataset. If a field is defined as an array or a set, then an index will be added to the variable name.
- **User Defined Type** name should not allow a number as the last character due to SAS limitations.

Hatteras™, RTI's Web-based CAI system, is used to speed-up development of Blaise instruments by generating files defining Blaise fields. Figure 1 below shows how expected field name and labels in SAS dataset are entered in the Hatteras™ SurveyEditor.

Figure 1. Entering SAS Field Name and Label in Hatteras™ SurveyEditor

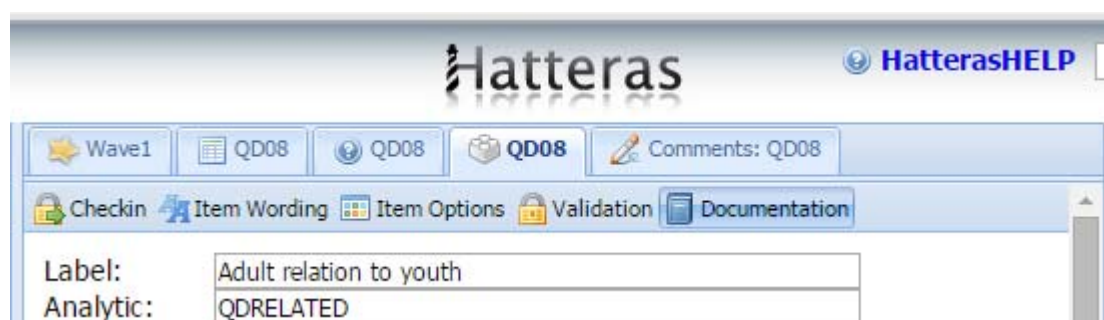


Figure 2. Example of Field Definition in Blaise Instrument

```
QD08 (QDRELATED)
ENUS
"HAND R SHOWCARD QD08.0/0/How are you related to ^prld.c_fname?"
/"Adult relation to youth"
: TREL
```

3.2. Initialization file

The application uses an initialization file (INI) to hold a few necessary settings in order to prepare the correct SAS input statements. If the name of the INI file is the same as the Blaise instrument name and exists in the folder where the BlzToSAS.exe is, it will be used by default. Otherwise, the name of the INI file should be provided interactively in a pop-up window.

There are two sections in the INI file. A section with [Project] name is required to specify common settings available for all studies. Section [Rename] is optional.

3.2.1 Common settings

Some common settings in the [Project] section of the INI file are required. Some can be omitted if the default values are acceptable.

- **ProjectID** – optional. It is needed if custom code is added to the application for the project.
- **Folder** – optional. Name of the folder where output files will be created, it could be entered interactively in a pop-up window before output files are created.
- **LabelLength** – required. If no label text is assigned to a question, then the first LabelLength characters of the question text are used as a label.
- **FieldNameLength** – required. The allowable length of field names varies by SAS version.
- **RemoveTabName** – optional. Default setting is “False”. If it is set to “True”, then table names will be removed from the field name by the program.
- **UseTypeNames** – optional. Default setting is “False” and it should be used if requirements to user defined types as described in 3.1 above are not in place. If the value is set to “True”, then SAS formats will use the names as defined in Blaise instrument. This is very helpful for creating SAS datasets for longitudinal studies to maintain names for the same formats across different waves of data collection.
- **UseTag** – optional. Default setting is “False”. If it is set to “True” and a tag exists for a field, then that tag will be used as a field name. For variable that is an array, the variable name of the array element will be created as a tag and “_” and the array index.
- **DK** – required. This value will be assigned to a variable if it has a value of “Don’t Know”.
- **RF** – required. This value will be assigned to a variable if it has a value of “Refusal”.

- **FileLabels** – optional. Not all variables can have labels defined during development of the instrument. This option is used when a file with missing labels becomes available.
- **FileRenames** – optional. Initial statements to remove block names can be entered as wildcard instructions in a file specified here or in the INI file in [Rename] section.

Example of the [Project] section in INI file with options described above.

```
[Project]
ProjectID = AAAA
LabelLength = 80
FieldNameLength = 30
RemoveTabName = True
UseTypeNames = True
UseTag = True
FileRenames = C:\AAAA\Inst_renames.txt
DK = -1
RF = -2
```

3.2.2 Rename Variables

Blaise instruments with embedded blocks and tables would have very long full field names and they cannot be used by SAS. A local Blaise field name is not always unique and can be defined as an array or a set. There are two ways that the BlzToSAS application can create output variable names suitable for SAS. Both of these options can be used in the same instrument:

- **Field Tag** is specified for a variable during development of the instrument.
- **Rename statements** are wildcard instructions specified in the INI file.

In addition, the following rules are used by default:

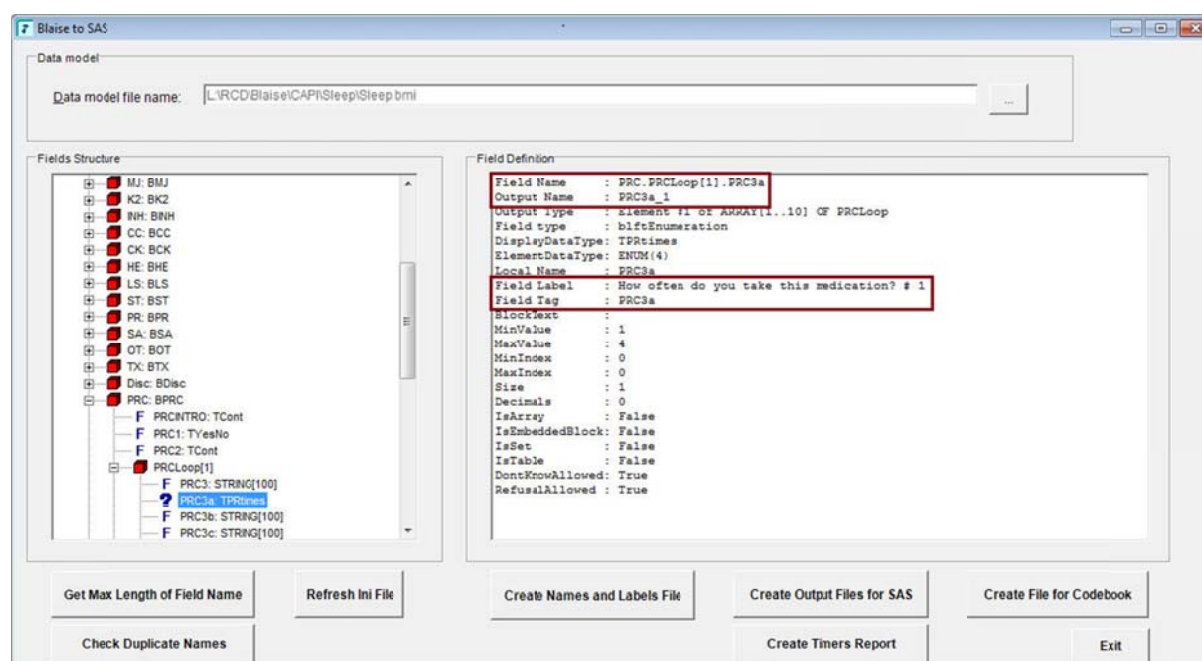
- Index representation for arrays of blocks is changed from “[i].” to “_i_”.
- Index for an array or a set field is changed from “[i]” to “_i” at the end of the output variable name.
- Dots between block names in output variable name are replaced by “_”.

Rename statements can be part of the [Rename] section in the INI file or can be entered in a standalone text file specified as “FileRenames” in the [Project] section. Use of the file with rename statements is more flexible and can start with the names of top level blocks that usually are not needed as part of the output variable name. For example, statement like (HHRoster.=) is used by the BlzToSAS application to create output variable names for all fields in “HHRoster” block without the block name. Other refinements can be added as the initial output is examined for possible improvements.

3.3 Review Variable and Type Names

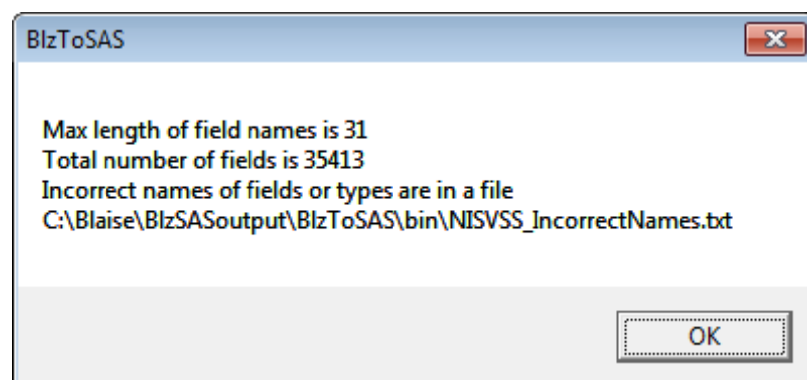
When the BlzToSAS application is launched and meta file of the instrument, along with the initialization file, are selected and loaded. The structure browser of the data model is displayed on the left panel and properties of any selected field in the tree are displayed on the right panel of the form. In addition to standard Blaise properties of an elementary field, “Output Name” and “Label” are presented as they will be defined in SAS dataset.

Figure 3. Blaise to SAS Form



The button “Get Max Length of Field Name” is used to examine the maximum length of output variable names and type names depending on the settings defined in the INI file. If there are some fields with lengths that exceed the allowable number of characters in expected output name, they are added to a file for review along with type names that cannot be used by SAS. It also shows the total number of fields in the Blaise instrument that will be outputted in an ASCII file. Depending on the available version of SAS, this can be used to decide if SAS dataset can be created for all of them.

Figure 4. Examine Blaise Instrument



After the file with long names has been reviewed, changes can be made to the INI file. Then the button “Refresh Ini file” is used to update settings used by the application. Now Blaise fields with long names can be selected in the tree and new output names will be displayed for them. Usually renaming is added for a block and the change appears for all fields in that block.

During this renaming process some of variables could be assigned a non-unique output name. The button “Check Duplicate Names” is used to find such variables and then the issue can be fixed by changing rename statements or in the Blaise instrument itself. If unacceptable type names are found, they can also be fixed in the instrument.

A few iterations of revisions to the INI file and the Blaise instrument might be needed before all problems are resolved to have variable and type names acceptable for SAS.

3.4 Review Labels

Labels for the SAS dataset can be defined in the field descriptions in the Blaise instrument during development if they are known at that time. Alternatively, labels can be revised by researchers who work with SAS datasets. An option was added to the application to prepare a file for them to review default labels for every variable in the Blaise instrument. The button “Create Names and Labels File” is used to prepare an Excel file with the following columns:

- Module name.
- Blaise full name of a variable.
- Blaise local variable name as it was defined in specifications.
- Output variable name for SAS.
- Default label (field description or specified number of characters from question text).
- Question text in default language.

When the spreadsheet is reviewed by researchers, they prepare an updated Excel spreadsheet with only two columns - output variable name and new label for that variable. This file is specified in the INI file and will be used by the BlzToSAS application to prepare labels statements for the output files.

3.5 Output files

To create the programs needed to create a SAS dataset from a Blaise database, two main programs are created by invoking button “Create Output Files for SAS”:

- Manipula setup to create an ASCII file.
- SAS program with include statements to create a SAS dataset from the ASCII file. Each include statement pulls in a major component to the SAS program such as label statements, format statements, and format associations with variables.

3.5.1 Manipula Setup

All fields from the Blaise database are exported by the Manipula setup generated by BlzToSAS. To simplify the process of recoding “Don’t Know” and “Refusal” to desired values specified in the INI file, code has been added to the setup for “string” fields allowing these special values.

Example of Manipula Setup.

```
SETTINGS
  DESCRIPTION = 'BLAISE to ASCII'
  INPUTPATH   = '\\...\DemoInst'
  OUTPUTPATH  = '\\...\BlzToSAS\ DemoInst '
USES
  Metal '\\...\ DemoInst\ DemoInst.bmi'
INPUTFILE InpFile: Metal (' DemoInst ', BLAISE)
SETTINGS
  ACCESS = SHARED
OUTPUTFILE OutFile: Metal (' DemoInst.asc', ASCII)

MANIPULATE
  IF InpFile.RV1.NOTES = DK THEN OutFile.RV1.NOTES := '-1' ENDIF
  IF InpFile.RV1.NOTES = RF THEN OutFile.RV1.NOTES := '-2' ENDIF

  OutFile.WRITE
```

3.5.2 SAS Program

The SAS program created by the BlzToSAS application utilizes include files to pull in input statements, labels, formats, format library references, and other options.

Example of SAS program:

```
libname library '\\...\BlzToSAS\DemoInst';
libname out '\\...\BlzToSAS\DemoInst';
TITLE 'DemoInst';
%include '\\...\BlzToSAS\DemoInst\DemoInst_procfmt.inc';
DATA out.DemoInst;
INFILE '\\...\BlzToSAS\DemoInst\DemoInst.asc' LRECL = 324290 TRUNCOVER;
%include '\\...\BlzToSAS\DemoInst\DemoInst_input.inc';
%include '\\...\BlzToSAS\DemoInst\DemoInst_toggles.inc';
%include '\\...\BlzToSAS\DemoInst\DemoInst_labels.inc';
%include '\\...\BlzToSAS\DemoInst\DemoInst_formats.inc';
%include '\\...\BlzToSAS\DemoInst\DemoInst_dk_rf.inc';
%include '\\...\BlzToSAS\DemoInst\DemoInst_drop.inc';
RUN;
```

Each “include” file in the program has its own purpose and can be extremely long depending on the number of variables in the SAS dataset:

- “**_procfm**” – SAS proc formats for enumerated types defined in Blaise instrument with additional options for “Don’t Know” and “Refusal” with values specified in the INI file.
- “**_input**” - Input SAS statements
- “**_toggles**” – SAS statements to define new recoded variables needed to convert “set” Blaise fields to toggle variables. The RTI convention for set questions is to recode them to a set of yes/no toggles and add the new toggle variables to the end of the dataset.
- “**_labels**” – statement to assign SAS variable labels for all variables including newly created recoded variables.
- “**_formats**” – SAS statement to assign format to all variables
- “**_dk_rf**” – SAS statements to assign “Don’t Know” and “Refusal” values for variables that were not changed by Manipula Setup in ASCII file.
- “**_drop**” – SAS statements to remove variables from the SAS dataset that have label “NoDeliv” in the Blaise instrument.

This SAS program can be run in a batch mode to create the SAS dataset from the ASCII file created after running Manipula setup. For Blaise instruments with thousands variables, the total number of lines in such a program becomes hundreds of thousands. When it is created by the BlzToSAS application, every variable that is expected to be present in the SAS dataset is there with the correct value, type, and label.

3.6 Timers Report

At RTI International, a standard procedure is used to calculate time spent during an interview in every section defined in the Blaise instrument. For interviews completed in more than one session, total time includes cumulative time for all sessions. The button “Create Timers Report” is used to create a SAS program that will produce the timers report.

Figure 5. Example of Timing Report

<p style="text-align: center;">Demolnst Interview Timing Report</p> <p style="text-align: center;">The MEANS Procedure</p>					
Variable	Label	N	Mean	Minimum	Maximum
Tmr_Interview_TimeTotal	Total Interview Time	18	99.42	15.75	187.85
Tmr_QD_TimeTotal	Core Demographics	18	2.53	1.26	6.32
Tmr_AS_TimeTotal	Child Cognitive Assessments	9	56.35	43.39	67.96
Tmr_RV1_TimeTotal	Rey Auditory-Verbal Learning Test (RAVLT 1)	9	1.16	0.51	2.00
Tmr_cal_TimeTotal	Calendar	9	6.94	5.43	9.63
Tmr_TUT_TimeTotal	ACASI Tutorial	8	5.83	1.92	13.63

The name of each timer section is defined as a field description for a timer section block.

4. Data Delivery

The application BlzToSAS is also used to help prepare final SAS datasets and a codebook after data collection for a study is completed. In addition, if special types of SAS datasets are needed, study specific buttons can be made available on the form.

4.1 Final SAS Dataset and Codebook

For data delivery, a codebook is often required to accompany each SAS dataset. The BlzToSAS application can create a simple codebook directly from the Blaise database by using the Blaise API. Alternatively, it can produce input for a more elaborate, SAS based, codebook generation system RTI has developed. This generic “Codebook Generator” is used for other types of instruments as well, including Hatteras™ Web and others packages.

The button “Create File for Codebook” is used to create a spreadsheet with metadata containing complete list of variables and the corresponding labels, format names, and question texts.

The SAS dataset produced by the BlzToSAS application, discussed in detail in section 3.5, usually needs to be altered for these situations to prepare final SAS datasets:

- Remove personally identifiable information (PII) data
- Remove variables used for intermediate analysis during data collection
- Add imputed or derived variables

Using the metadata spreadsheet, project staff can add additional information not available from the Blaise instrument or revise the labels and question texts as they might contain formatters and/or fill variables. Project staff would also decide which variables to keep or drop in the final dataset. Example of the spreadsheet is shown in figure 6 below.

Figure 6. Metadata Spreadsheet

	A	B	C	D	E	F	G
1	SAS_Variable_Name	Length	Var_Order	Label_of_Variable	SAS_Forma	Drop_Keep	QuestionText
2	MC_zrid	8	1	Case ID	\$ch2fmt	Keep	Case ID
3	MC_sum_evt	4	2	Current Event Code		drop	Current Event Code
4	MC_sum_evtxt	15	3	Description of current event code		drop	Description of current event code
5	MC_sum_stat	4	4	Summary Status of case	\$statscr	Keep	Summary Status of case
6	MC_sum_sttxt	15	5	Summary Status of case - text	\$ch2fmt	Keep	Summary Status of case - text

Once the review process is completed, the SAS dataset and metadata spreadsheet become the input to the codebook generation process consisting of a set of SAS programs. The main steps in this process are:

1. Read spreadsheet metadata
2. Generate SAS scripts to
 - a. Retain/drop variables
 - b. Rename variables
 - c. Assign labels and formats
3. Read SAS dataset, apply the above scripts to create the final dataset
4. Generate dataset contents and frequencies from the final dataset
5. Create codebook document

The codebook uses HTML structure and is styled using cascading style sheet (CSS). It can be delivered as is or brought into Word for additional editing, such as adding headers, footers, or creating 2-column format. Example of a sample codebook is shown in figure 7 below.

Figure 7. Codebook Example

Screener Codebook September 9, 2013					
<hr/>					
Variable:	Person_1_SCQ_11				
Description:	Is HH member 1 eligible				
Stem:	HHMEMBER				
Notes:					
Label	Value	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Legitimate skip	.	3	1.52%	3	1.52%
YES	1	198	98.48%	198	100.00%
DELETE	4	0	0.00%	198	100.00%
Dontknow	-1	0	0.00%	198	100.00%
Refusal	-2	0	0.00%	198	100.00%

Variable:	Person_2_ageR				
Description:	Age of person 2				
Stem:	How old is "firstname"? ENTER 0 FOR LESS THAN 1 YEAR				
Notes:					
Label	Value	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Legitimate skip	.	58	29.29%	58	29.29%
Data present		137	69.19%	195	98.48%
Dontknow	-1	2	1.01%	197	99.49%
Refused	-2	1	0.51%	198	100.00%

4.2 Special SAS Datasets

The standard SAS dataset has one record per respondent. For studies collecting information about household members or people with whom respondents interact, the same questions are asked in loops. The data is collected in tables or arrays and for analysis it can be grouped the same way as it was collected. For example, for one respondent there can be blocks of questions for each household member. Clients sometimes prefer reorganization of this type of data into separate relational datasets.

The BlzToSAS application uses BCP to simplify this process of creating relational SAS datasets when they are requested by the study. A list of common variables and blocks with the required data must be defined. Then custom code can be added to the application to process them. Standard procedures are used to define variable names, formats, labels, and input statements along with the SAS program. The time needed to add custom code to the application is significantly less than creating a comparable SAS program from scratch.

To create special SAS datasets, the Blaise database name should be provided to create two files with data – an Excel spreadsheet and an ASCII text file. The Excel spreadsheet can be used to easily check the output data.

5. Conclusion

SAS has become the standard for file deliveries at RTI International. Projects producing extremely large data models or data models with complex nested structures and/or relational structures benefit from the enhanced system provided by BlzToSAS for generating SAS datasets. RTI's adoption of

this system has paid off in efficiency gains and improvements in the quality and reliability of file deliveries. It has also been well integrated into our new codebook generation system, so that a visually appealing codebook document can accompany our file deliveries.

6. References

Blaise 4.8 API Reference Manual

Roger Schou, “Creating Code to Convert Blaise to ASCII for SAS Input (Using the Blaise API within Visual Basic)”, 2003, Proceedings of 8th International Blaise Users Conference

Barbara S. Bibb & R. Suresh, “Automatic Customization of Datasets Using Cameleon”, 2004, Proceedings of 9th International Blaise Users Conference

An Improvement of the Application of Blaise CAWI in Verifying Systems of CFPS

Shuang Yu, Yongjian Zhang, Jiahui Yao, Peking University

Abstract

Instant listening recording function was used in our recording verification questionnaire of the verification system in 2014. It is similar to the ACASI's convenient recording loading, but also facilitates the inspectors not have to pick the recording which need to listen. It saved much time.

Our verification questionnaire gets fixed recording fields. After CARI extracted audio file and the recording fields is loaded in the verification questionnaire, inspectors can listen to the recording for verification when they meet such fixed fields.

But we also encountered a number of problems in ACASI, such as multiple recording file generated by the CARI for a single field, so that we're not sure which to load is the correct and effective recording file.

1. Background

Chinese Family Panel Studies (CFPS) is aimed at three levels data collection, including individuals, families and communities. The project studies the changes of society, economy, population, education and health, which is reflected by the collected data. It is a key social science project that serves academic research and policy decision supporting. The first phase of the survey is 12 years (2008-2020). CFPS is early funded by phase II of 985 plan of the Ministry of Education, beginning preparations in 2005, setting up an institution in 2006. In 2007, preliminary work completed, including two test surveys, 140 families totally, in Beijing, Hebei, Shanghai, and the questionnaire tools basically mature. Exploratory survey in 2008 in Beijing, Shanghai, Guangdong, expanded scale to 2400 families. Tool testing follow-up survey in 2009 in Beijing, Shanghai, Guangdong, tested the stability and reliability of CAPI technology, real-time management techniques for survey process, real-time technical support system for survey process, real-time data quality monitoring techniques. In 2010, the survey all over China (Tibet, Qinghai, Xinjiang, Ningxia, Inner Mongolia, Hainan, Hong Kong, Macao and Taiwan are not listed) was formally implemented, the scale was up to 16,000 families, and a follow-up survey was implemented annually.

Every year from 2010, field surveys are being conducted from March to July, data is being cleaned up from August to October, and from November to February of next year, provide the cleaned data for the teachers and students in Peking University and partner universities, write the key indicators report "China Report", which is published by Peking

University Press. The key indicators report and data is published to public in March. [1]

There are 4 kinds of questionnaire in CFPS, including community, family, adult and child questionnaire. The community questionnaire is used to obtain the household residence of the basic environmental information; family questionnaire is used to obtain family economic, demographic status information; adult and child questionnaire is used to obtain the individual social, economic, educational and health information.

2. The insufficiency of traditional verification

In order to ensure the quality of data, and to learn about supervise the interview process, the project group needs to record the interview process. For the recording of the interview process, the verification team have dedicated personnel for verification in order to ensure the quality of survey. Therefore, the project group will prepare the verification questionnaire to verify and supervise interview process. The result of verification is among alternative answers, subjective answers, jumping with shortcut, questioning incomplete, not meeting the meaning, and detailed-questioning insufficiently.

Traditional methods of verification are applying Blaise CAWI. The inspector answers to the survey questions of verification, and at the same time open the audio playing program to verify recording situations. When the inspector listen to the record audio, he or she should search by module, by sample number in advance. As shown in Figure 1, the original verification system search for recording audio. This approach requires a lot of manual intervention. It is also prone to problems that sample does not match.



Figure 1

3. The solution

According to the previously described problems, according to ACASI design ideas, ISSS proposes appropriate solutions. Its general scheme is to integrate the verification questionnaire of CAWI with audio playing program's capabilities.

In 2014, the verification questionnaires implement the scheme that recording audio files will play automatically when the inspector verifying. Firstly, embed the recording audio file in every verification question. Secondly, applying the new HTML5 element on the page, the system automatically plays the recording audio file. It help verifiers saving find the time to find the exact recording audio file, which greatly saves the verification time. It improves the efficiency of the verification.

The detailed steps are as following:

1. Define a variable to store the path of the recording audio file. For example, as shown in Figure 1, the file path is combined with other three variables: interviewerid, countryid and sampleid, and is assigned to the variable FamilyLY.

```
FamilyLY:='HTTP://10.8.8.73/WAVFiles/'+interviewerid+'/'+countryid+'/'+ SampleID
```

Figure 1. Define a variable for path.

2. Define a variable to store the name of the recording audio file. As shown in Figure2, it is the combined with sample ID and question ID.

```
XA101a:= SampleID+'SECA.TA1'
```

Figure 2. Define a variable for name

3. Embed the recording audio file path and name in the verification questions. As shown in Figure 3, the variables as defined in step 1 step 2 are assigned to the property src, the height and width is set according to the player layout. This is the application of a new HTML5 element: audio. The element can define the sound, such as music or other audio streams. Its properties are introduced in Table 1.

```
A101a(A101a)"@BA101a@B 访员是否提问了请您确认 ^preload.fmiwtime_last 调查时您家是否有以下
人员？ ^preload.FML 一题相关内容？ @/(题号：TA1 录音名：SECA.TA1)
@/@/@#<audio controls autoplay>
<source src=""^FamilyLY/^XA101a.wma"" type=""audio/mpeg"">
<embed height=""50"" width=""100"" autoplay=""false"" autostart=""false"" src=""^FamilyLY/^XA101a.wma"">
</audio>@# ":TContinue
```

Figure 3. Embed an audio file to a question

Table 1. Properties of element audio

Name	Value	Description
autoplay	autoplay	Audio plays automatically.


controls	controls	Display the playing controls.
loop	loop	Audio plays in loop.
muted	muted	Audio will be muted.
preload	preload	Audio will be preloaded when the page is loaded. And it will be ignored when autoplay property is set to autoplay.
src	url	Audio URL path.

4. After the completion of these three steps, and the published verification questionnaires will see as shown in Figure 4.

In addition, as a question because many times to be asked will produce multiple recording audio files, and all these audio files need to be heard, it is not convenient. So we developed an audio merger program with C#. The program can automatically find all recording audio files of a question, and then all these files are combined into a single audio file. Finally the combined audio files are embedded in the verification questionnaire.

After a series of improvements, with the ACASI mode is used, the audio files are embedded directly in to the question of verification. When a verification questionnaire is processing, the audio files plays automatically. It saves time, increases efficiency, changes audio verification mode.

D107a 访员是否问过“包括和家人外出吃饭，一般情况下，您每周有几个晚上会和家人一起吃饭？”一题相关内容？
(题号：Q1002录音名：SECQ.QQ1002)



☒ 1. 是
 ☐ 5. 否
 ☐ 9. 无法判断
 ☐ Don't know
 ☐ Rather not answer

Figure 4. Verification sample page

4. Summary

ISSS Peking University applied ACASI idea in CFPS verification in 2014, integrated the audio file into the verification questionnaire. The practice has proved that this method in 2014, has played a significant role in the verification process. Inspectors are no longer used to find

audio files, can focus on verifying each problem, improve the efficiency of the verification, and improve the quality of verification, actively and effectively promote the development of verification.

Prototyping Touchscreen Blaise Applications

Karl Dinkelmann, Heidi Guyer, & Rebecca Gatward, University of Michigan

1. Introduction

The integration of new technology in survey research has led to new developments in computer-assisted interviewing applications. Touchscreen functionality can improve usability and, potentially, improve data quality and reduce interviewing costs by providing a more user-friendly, straightforward means of navigating through a data collection instrument. With several large-scale, in-person data collection screening studies on the horizon at the University of Michigan, and plans to purchase new hardware, we have explored existing touchscreen capabilities within Blaise 4.8. In this paper, we will describe the development of a touchscreen Blaise screening application prototype in consideration for two studies that will begin data collection in 2015-2016. The developmental goal of this exploration is to design a touchscreen household screening application to be used by field interviewers to screen over 50,000 households in the next year. The prototype builds on the work done by Hofman and colleagues in 2013 and furthers the development of touchscreen applications in Blaise 4.8.

2. Background

Survey Research Operations (SRO), a unit within the University of Michigan's Institute for Social Research's Survey Research Center, has used Fujitsu Lifebook Convertible Tablet PCs for close to a decade. Between 2005 and 2010, the Fujitsu Lifebook T4220 utilized Windows XP Tablet PC Edition and was limited to a digital pen (or stylus) input while in tablet mode. The laptop pool was refreshed in 2010 with the Fujitsu Lifebook T730's which used Windows Enterprise 7 64 bit and had dual digitizers (digital pen input and multi-touch screens). In 2015, we began replacing the laptop pool again with Fujitsu Lifebook T734's. The T734 came with the option of having Windows 8.1 or 7 64 bit editions. The Survey Research Center has decided to forego upgrading to Windows 8 with the intention of moving to Windows 10 instead. Therefore, the T734's are loaded with Windows Enterprise 7 64 bit with dual digitizers. Despite having the digital pen input option available for ten years and gaining touchscreen ability five years ago, these features have only been used to a limited degree in a production environment. In fact, the touchscreen functionality was disabled in the production environment on the T730 pool of laptops (2010-2015).

Several proofs of concept were created and explored in an attempt to use a digital pen as a means of input. Examples include exploring the possibility of using Blaise Basil for ACASI instruments. The interest in Basil resulted from its ability to provide developers greater control of screen design. However, it was found to be significantly more difficult to program. Spaulding (2012) demonstrated the creation of a touchscreen ACASI instrument using Basil developed by the US Census Bureau for the National Health Interview Survey. The Blaise 4.8 Help file describes Basil as,

“...an alternative data entry program for Blaise questionnaires, specifically designed to handle Self Interviewing situations. BASIL combines the strength of the Blaise rules engine with the flexibility of Maniplus, which allows programmers of BASIL to create tailor-made applications.”

While SRO did not adopt Basil for ACASI instruments, an application was developed to capture respondent signatures during the informed consent process and to sign cash receipts using the stylus. This process was recently updated and expanded in Blaise (Gatward et. al. 2015).

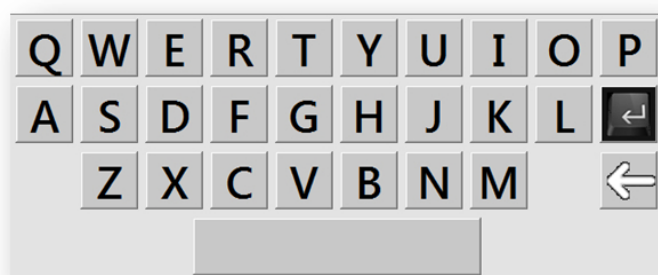
The limited use of the digital pen and touchscreen features to date is primarily due to the lack of features integrated within our core set of supported software products used to support the collection of survey data in the field until now.

3. From Keyboard & Mouse to Touch-Friendly Applications

The process of transitioning from traditional keyboard and mouse driven data entry to one that is touch-friendly can take many forms and involves understanding the core elements needed for one to complete a given task, or set of tasks, without the use of the traditional mouse or keyboard. If the task requires a keyboard entry, is the operating system's native touchscreen keyboard sufficient? If the native keyboard does not work, then one must build their own keyboard. Finally, the traditional menu driven process within software applications was not created with touch input in mind. Therefore, this transitional process needs to take into account larger elements allowing for finger input, as is done via a keyboard.

Our prototype took a transitional process that built on successive approximations and the starting point was a sample menu provided by Statistics Netherlands, used in the Caribbean Netherlands Implementation (Hofman et. al. 2013), and progressed through fluid iterative processes of exploring what was possible. A key requirement was the ability to enter text into the application at the point where we capture individual's names within a household listing while screening for eligible respondents. We created many keyboards and the final proof of concept implemented was to emulate the look and feel of a more familiar mobile keyboard within the Blaise Menu file. This led to building an elaborate keyboard using the Blaise Menu file. Building a new keyboard required several iterations in order to understand the placement of elements. An early version of the keyboard prototype is presented in figure 1. Although, this version would allow one to enter an individual's name on text fields in an instrument, this standard keyboard was not ideal because of the rudimentary nature of the keyboard.

Figure 1: Example Keyboard in a Blaise 4.8 Menu File



We then began to look for available images that would provide a cleaner presentation that appeared more like a real keyboard. One version had black keys with matte letters, as shown in figure 2. This keyboard is completely functional and gives the feel of a more authentic keyboard presentation. Nevertheless, we felt we could do better.

Figure 2: Keyboard in a Blaise 4.8 Menu File using Images of Black Keys



Despite the availability of these two keyboards, we wanted to attempt to emulate a keyboard design that was cleaner and decided to attempt to replicate a design more widely used on mobile platforms in the Blaise Menu file. A screenshot of the mobile platform keyboard is shown in figure 3 and figure 4 shows the Blaise menu file attempt to emulate it.

Figure 3: Mobile Keyboard



Figure 4: Blaise Menu File Emulating a Mobile Keyboard



The cleaner keyboard images, as displayed in figures 3 and 4, sparked enthusiasm in the Blaise programmer in the potential options that could be realized in developing a new on-screen keyboard. However, keyboard images readily available had limitations. Not all of the keys were always present and the simple, clean look that we were looking for was not always available. Therefore, it was at this point that we realized that we would need to begin to create the desired keys and keyboard from scratch. Developing a new keyboard would involve creating keyboards in a graphical creation software program such as Adobe Photoshop. We used the diagramming and vector graphics applications within Microsoft Office Visio. We used Visio to build the raw keyboards and then increased the presentation of the keyboards to 400% to take accurate screen captures, key by key. The larger screen captures were later scaled into the appropriate pixel width and height for the given item within the Blaise menu file. For example, the button pixel dimension determined for the main menu panel were 56 by 56 pixels and the typical keyboard keys were scaled to the pixel width and height of 70 by 45.

Ultimately, this process of building the keys and buttons from scratch provided the desired method to produce a screen presentation that was simple and clean. However, it is likely not the quickest method. A graphics designer rendering a series of key images in advance could enhance the process of designing a keyboard.

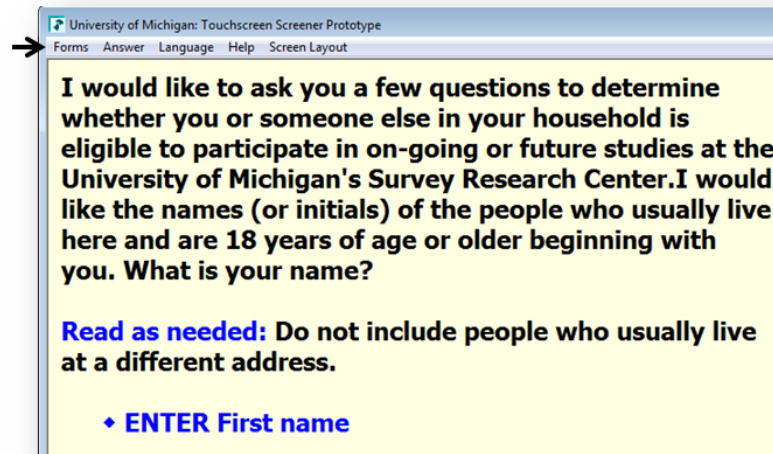
4. Blaise Touchscreen Implementation Essentials

Once we began to explore the capabilities added to the Blaise Menu file, it became clear that a potential solution could exist for our first production touchscreen application. Specifically, Blaise 4.8 added the capability of enhancing the menu file, by replacing the button bar concept with a panel concept (Blaise 4.8 Online Help, 2015). This was done to allow for greater flexibility in Blaise IS. However, these features were also exposed within the Data Entry Program. In this section, we discuss three Blaise elements essential to this touchscreen prototype: the menu file, the mode library settings, and menu data storage.

4.1 Menu File

Blaise 4.8 introduced a “panel concept” within the Menu file offering one the ability to create a completely different means of presenting the traditional menu options on the button bar. For the UM Blaise 4.8 touchscreen prototype, the majority of the work was completed in the Menu file within the Blaise Menu Editor. The traditional Blaise menu presentation is shown in figure 5:

Figure 5: Traditional Blaise Menu File



The key elements of the menu file, in our opinion, are the panels and their controls (child elements of panels), visibility expressions, and control actions. The uses of these elements as they relate to the touchscreen prototype are discussed further.

4.1.1 Panels and Controls

Panels are the parent and controls are the child of a panel. Attributes defined in the panel can be inherited or overwritten in the control. For example, the panel is typically a set of items presented in one row or column and the controls are the items that are in the panels. Figure 6 shows the bottom row of a simple keyboard designed for landscape presentation. Figure 7 shows all of the controls within this panel.

Figure 6: Example Panel in the Prototype

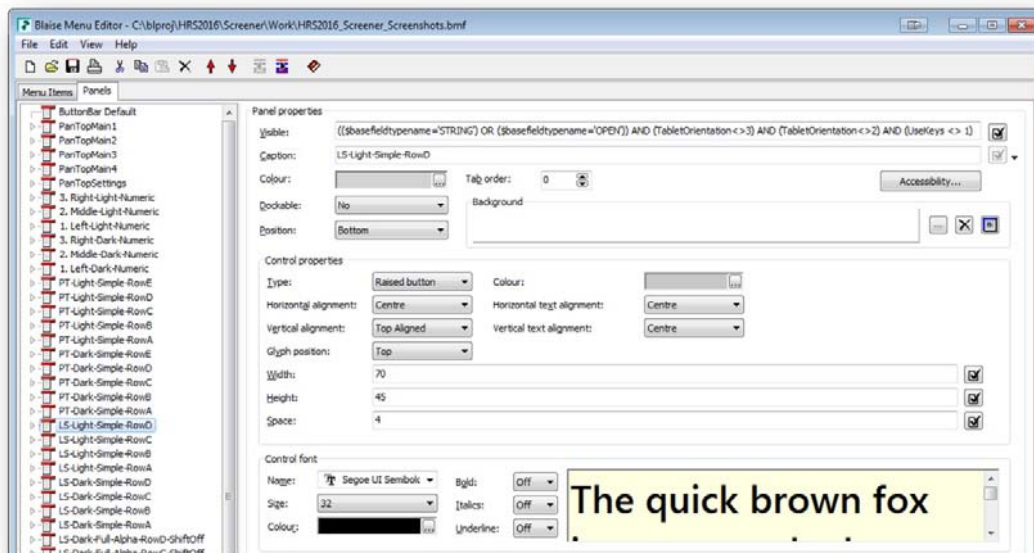
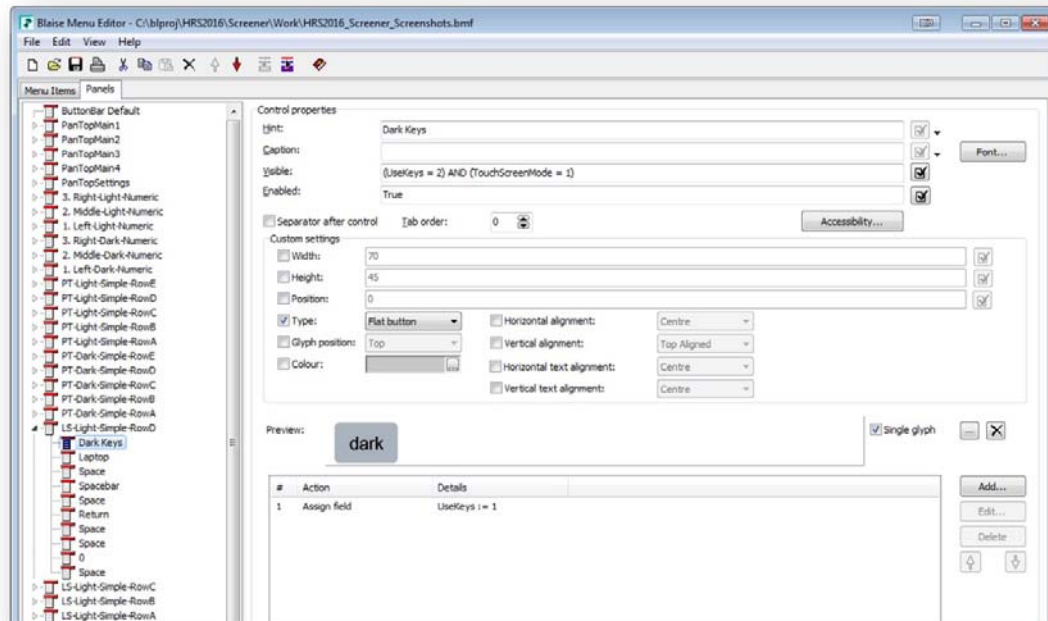


Figure 7: Example Control in the Prototype



Each row or column of a given element in a touchscreen prototype is represented as a panel on the left. Therefore, if an alphanumeric keyboard consists of four rows, there are four panels for the keyboard in the Menu file. Previously, in section 3, we discussed the methods we used to conceptualize and construct the keyboards into the menu file.

4.1.2 Visibility Expressions

Using expressions within visibility conditions of panels and controls is powerful. This allowed for great flexibility when building the touchscreen prototype. Expressions allowed us to build one menu file containing many different keyboard presentations. For example, if a question had “don’t know” or “refused” attributes we could use expressions to display the “don’t know” or “refused” buttons. Otherwise, those buttons are not displayed for a given question. We also used data stored within the data model to assist in driving the presentation of items on the screen; this is discussed further in section 4.3 Menu Data Storage.

It is important to understand if one expression fails within menu files it has a cascading effect that will prevent many of the menu panels from displaying properly. Therefore, it is important to meticulously test expression usage within menu panels and controls.

4.1.3 Control Actions

Control actions are the assignments attached to a given control. For example, the “A” key has a ‘keyboard action’ of the key value of [A]. During our review, we attempted to replicate all actions of a full keyboard and were able to map everything with the exception of a double quote key. Other items we could not map were the Euro “€”, Pound “£”, and Yen “¥” signs. However, this could have been because our U.S. based keyboard did not have the options for these keys. Control actions are important, as they become the actions and assignments behind the scenes that drive the touchscreen presentation and navigation through the instrument.

4.2 Mode Library Extensions

Two new layout sets were added to the mode library file and the normal laptop CAPI layout set was retained as well. The two additional layout sets added were for portrait & landscape touchscreen presentation on the tablet. The laptop set allows us to flip back to our traditional CAPI presentation. The

portrait layout set maximizes the tall and narrow screen presentation. The landscape layout set uses the width of the screen to maximize the placement of the various elements.

Other mode library changes were to increase the default font to 16-pt. and make it bold. Within the portrait and landscape layout sets, we changed the answer list button sizes to 13 and changed all of the backgrounds of the InfoPanels to white. A white background provides better contrast which is important in outdoor or daylight settings, as will be the case when screening households using a touchscreen application in the field.

4.3 Menu Data Storage

While creating the prototype, we needed a means of storing current settings and other actions associated with menu options and the control assignments to dynamically alter the menu presentation. In Blaise, this can be done using registry settings and environment variables in Windows. However, given the rapid turnaround of the prototyping process we created a series of constructed variables in the data model that were always on route and had .KEEP assignments, allowing for successive iterations where we could easily build data values from these variables into the visibility conditions. Figure 8 shows a table of the constructed variable we built into the data model to aid in driving the dynamic presentation of the menu panels. Many of these variables were added to aid in the demonstration of the prototype and are likely not needed in a production setting.

Figure 8: Constructed Variables Used in the Prototype

Constructed Variable	Description	Values (Default = bold)
UseTouchscreenMenu	Determines if the touchscreen menu can be used or not	1 = Yes 5 = No
AllowMods	Determines if a user can make changes to the current touchscreen settings	1 = Yes 5 = No
TabletOrientation	Tablet screen orientation	1 = Landscape 2 = Portrait 3 = Laptop Mode
TabletMode	Tablet Mode	1 = Laptop 2 = Tablet
UseKeys	Which Keys to use	1 = Dark Keys 2 = Light Keys
Keyboards	Which Keyboard is selected	1 = Basic/Simple 2 = Full: Mobile Alpha 3 = Full: Mobile Numeric 4 = Full: Mobile Symbol 5 = Full: Win7 OSK Alphanumeric
OS_KeyStyles	Which style should be used	1 = Mobile 2 = Win7
L10N_Keyboards	Use a localized version of the keyboard, US Qwerty only option	1 = US Qwerty 2 = <i>UK</i> 3 = <i>Chinese</i> 4 = <i>Nepali</i>
LangSelected	If more than one language is used, which language has been selected	1 = English 2 = Spanish
TouchScreenMode	Determines how and what is presented to the end user at run time	1 = Demo 2 = Testing 5 = Production
ChangeSettings	Opens the settings touchscreen toolbar	1 = Show settings touchscreen bar 5 = Touchscreen Settings off
ExitSettings	Use to exit the settings touchscreen toolbar	Empty 1 = Yes
AllowCAPI	Determines if the normal CAPI mode is an available option	1 = Yes 5 = No
ShiftState	If the shift button has been selected or not	1 = Inactive 2 = Active
CapsLockState	If the cap locks button has been selected or not	1 = Inactive 2 = Active












5. UM Touchscreen Prototype

The UM touchscreen prototype is made up of a touchscreen toolbar, alphanumeric keyboards and numeric keypads depending on the underlying question type. Other elements on the screen have been made larger for a touch-friendly user experience.

5.1 Touchscreen Toolbar

The touchscreen toolbar replaces the traditional Blaise menu (seen previously in figure 5). It is made up of four menu panels where images change to give the illusion that one panel is being used. This can be accomplished by using one panel and defining the fixed pixel location of elements and their associated images. However, the multiple panel method was chosen for its ease of scaling across multiple tablets that have different pixel density because of different screen resolutions and sizes. The location of two buttons changes between four images in the four panels. The first button location that changes is the landscape and portrait options which allow the user to switch between the landscape and portrait layout sets depending on the current tablet orientation. The second button location is the language selection button where one can switch between English and Spanish languages in the prototype. Figure 9 provides a matrix of the touchscreen toolbars and how the visibility conditional on the panels and controls interact with associated actions and assignments of these four panels to give a single unified appearance.

Figure 9: Touchscreen Toolbar Visibility, Action, and Assignment Matrix

Image	Panel				Action & Assignment	Visibility Condition
	PanTop-Main1	PanTop-Main2	PanTop-Main3	PanTop-Main4		
	Visibility Conditions:					
	TabletMode=2 AND ChangeSettings <> 1					
	TabletOrientation<>2		TabletOrientation=2			
	LangSelected = 2	LangSelected = 1	LangSelected = 2	LangSelected = 1		
Control						
	Previous				Pervious Field	True
	Portrait				Assign field TabletOrientation := 2 TabletMode := 2 Set layout set = Portrait	Determined in the panel
			Landscape		Assign field TabletOrientation := 1 TabletMode := 2 Set layout set = Landscape	Determined in the panel
	F1				Question Help	(\$LOCALFIELDTAG='Help')
	F2				Make Remark	True
	Don't Know				Don't Know	position('D',\$ATTRIBUTES)>0
	Refusal				Refuse	position('R',\$ATTRIBUTES)>0
	Settings				Assign field ChangeSettings = 1	AllowMods = 1
	English		English		Assign field LangSelected := 1 Set Language = ENG	Determined in the panel
		Spanish		Spanish	Assign field LangSelected:= 2 Set Language = SPA	Determined in the panel
	Next				Next Field	True

5.2 Alphanumeric Keyboards and Numeric Keypads

The current prototype consists of a series of alphanumeric keyboards and numeric keypads to demonstrate potential features that could be leveraged in a touchscreen application. Additionally, we felt it was necessary for the interviewer to be able to dynamically switch between light and dark keys. The primary concern here was due to unknown external lighting factors associated with screening both indoors and out, as well as the influence of weather conditions. The main, 'simple', keyboards were created as a proof

of concept. Figures 10a and 10b show examples of the simple keyboard with light keys and figures 11a and 11b show a similar keyboard presented with dark keys.

Figure 10a: Text Entry w/Light Keys (Landscape Presentation)

University of Michigan Touchscreen Screener Prototype

Navigation icons: back, home, F1, F2, help, warning, Sp, forward.

I would like to ask you a few questions to determine whether you or someone else in your household is eligible to participate in on-going or future studies at the University of Michigan's Survey Research Center. I would like the names (or initials) of the people who usually live here and are 18 years of age or older beginning with you. What is your name?

Read as needed: Do not include people who usually live at a different address.

• ENTER First name

Add HH Members	First name	Last name	Age

Keyboard layout (light keys):

- Row 1: Q, W, E, R, T, Y, U, I, O, P, 1, 2, 3
- Row 2: A, S, D, F, G, H, J, K, L, 4, 5, 6
- Row 3: Z, X, C, V, B, N, M, [backspace], 7, 8, 9
- Row 4: [dark mode toggle], space, [return], 0

Footer: HHNames.HH_Names[1], V155APhone, 403346, 3/18/2015, 1:05:49 PM, ENG

Figure 10b: Numeric Entry w/Light Keys (Landscape Presentation)

University of Michigan Touchscreen Screener Prototype

Navigation icons: back, home, F2, help, warning, Sp, forward.

And what is your age?

If INF cannot or will not provide answer, first probe for the year of birth and subtract that from the current year to find the age.

If you still cannot get an answer, ask the INF if the person is 40 or older. If no, enter [997].

If yes or if INF cannot or will not say, exit the screener and find a different INF.

Add HH Members	First name	Last name	Age
	FIRST NAME 1	LAST NAME 1	

Keyboard layout (light keys):

- Row 1: 1, 2, 3
- Row 2: 4, 5, 6
- Row 3: 7, 8, 9
- Row 4: 0, [return], [dark mode toggle]

Footer: HHNames.HH_Names[1], V1557_Age, 403346, 3/18/2015, 1:06:29 PM, ENG

Figure 11a: Text Entry w/Dark Keys (Landscape Presentation)

University of Michigan Touchscreen Screener Prototype

I would like to ask you a few questions to determine whether you or someone else in your household is eligible to participate in on-going or future studies at the University of Michigan's Survey Research Center. I would like the names (or initials) of the people who usually live here and are 18 years of age or older beginning with you. What is your name?

Read as needed: Do not include people who usually live at a different address.

♦ ENTER First name

Add HH Members First name Last name Age

Q W E R T Y U I O P 1 2 3
A S D F G H J K L 4 5 6
Z X C V B N M ← 7 8 9
light space return full 0 exit

HHNames.PH_Names[1].X1058A?Name 4/22/16 3/18/2015 1:06:07 PM ENG

Figure 11b: Numeric Entry w/Dark Keys (Landscape Presentation)

University of Michigan Touchscreen Screener Prototype

And what is your age?

If INF cannot or will not provide answer, first probe for the year of birth and subtract that from the current year to find the age.

If you still cannot get an answer, ask the INF if the person is 40 or older. If no, enter [997].

If yes or if INF cannot or will not say, exit the screener and find a different INF.

Add HH Members First name Last name Age

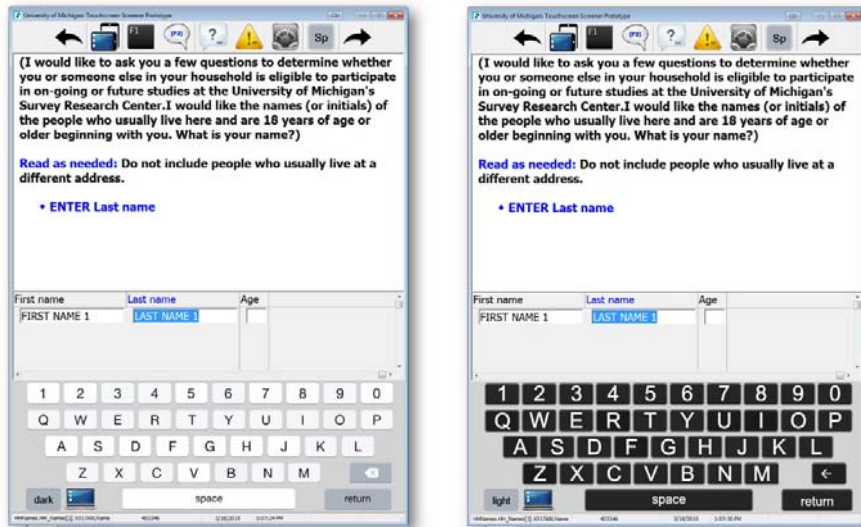
FIRST NAME 1 LAST NAME 1

← 1 2 3
4 5 6
7 8 9
0 enter
light exit

HHNames.PH_Names[1].X1057_A?Age 4/22/16 3/18/2015 1:06:25 PM ENG

In addition to offering keyboards for text keypads for numeric while in the landscape orientation we also added a different keyboard for text entry where the numbers moved on top of the keys. Figure 12 show how this would look if one were viewing the tablet in portrait orientation.

Figure 12: Text Entry in Portrait Orientation



The UM Touchscreen Prototype was expanded to explore whether a full keyboard, including punctuation and arrow keys, was possible. Two models were created to evaluate if a full keyboard could be created with the available custom keyboard actions with the actions associated with the available keyboard actions in Blaise. The two models for these different keyboards were created using dark keys to mimic the Windows Tablet Input Panel (figures 13a and 13b) and style presented in a mobile keyboard (figure 14a, 14b, and 14c).

Figure 13a: Windows OSK Style Full Keyboard w/Shift Inactive



Figure 13a: Windows OSK Style Full Keyboard w/Shift Active



Figure 14a: Mobile Style Full Alpha Keyboard (w/Shift Active)



Figure 14b: Mobile Style Full Numeric Keyboard



Figure 14c: Mobile Style Full Symbol Characters Keyboard



6. Future Exploration

Our preliminary findings suggest that Blaise 4.8 can leverage many of the desired touchscreen features to meet our current needs. Nevertheless, our next step will be to evaluate similar and applicable features, as they exist within Blaise 5. As it was not possible to accomplish similar touchscreen features with Blaise 5 when the majority of the touchscreen prototyping work was done and many questions linger. Will it be possible to build a similar onscreen keyboard within the Windows Data Entry Program in Blaise 5? Perhaps the iOS or Android Blaise 5 applications will alleviate some of these issues.

7. Summary

As touch-friendly devices become ubiquitous and proliferate the landscape, we can only hope that many of these extra measures to develop useable touchscreen applications become obsolete. Until then, the Blaise menu panel approach to extending Blaise 4.8 instruments to touchscreen represents exciting possibilities moving forward. A significant plus to this method is that it can be created once and reused on multiple instruments with minimal source code changes. We feel the features explored here open up exciting possibilities for the Blaise user community and the possibilities seem limited to one's own imagination!

8. References

Blaise 4.8 Online Help (Version 4.8.4.1899). (2015).

Gatward, R., Dinkelmann, K., Ogden, I., Dworak, P., & Guyer, H. (2015). *Converting the administration of health measures and bio specimen collection from paper to CAI*. The Sixteenth International Blaise Users Conference (IBUC), Beijing, China.

Hofman, L., Dolmans, R., & de Bolster, G. (2013). *Blaise on a Windows 8 tablet. The Caribbean Netherlands Implementation*. The Fifteenth International Blaise Users Conference (IBUC), Washington, DC., USA.

Spaulding, T. (2012). *Implementing Audio Computer Assisted Self Interviewing in Basil*. The Fourteenth International Blaise Users Conference (IBUC), London, UK.

Converting the Labour Force Survey to an electronic questionnaire using Blaise 5

Mike Hart, ONS

The initial Electronic Data Collection focus for social surveys is the Labour Force Survey, a choice driven by the Labour Force's importance, size, cost, longitudinal design, and opportunities to collaborate with other countries that are also focusing on web data collection research on their LFS.

The aim is to introduce a self-completion web mode to cover part of the sample, thus moving to a mix of three data collection modes: web self-completion, face-to-face interview and telephone interview.

Potential benefits from introducing a web mode include:

- Reduce data collection costs, by decreasing the number of interviews
- Meet respondent expectations i.e. the demand for web / digital services
- Stem falling response rates e.g. by providing flexibility to respondents, who can self-complete the survey at a time convenient to them (within the survey period)
- Drive design improvements across all collection modes
- Reach groups who won't respond via other modes i.e. if specifically targeted via web (although research indicates untargeted web surveys generally achieve similar socio-demographic coverage to existing modes)

The extent to which these benefits can be delivered will be determined during the course of the research. It is unlikely that we can achieve a survey design that delivers all of these benefits i.e. reduce costs, change sample demographics and increase response – each would require a different design. Furthermore, the survey's complexity (operations, survey design, questionnaire length and convolution) plus the high profile nature of its outputs, makes this a significant challenge. This challenge requires major investment and a considered and logical research approach, with a high degree of caution to ensure that the quality of statistical outputs is controlled and managed and that operational stability is maintained.

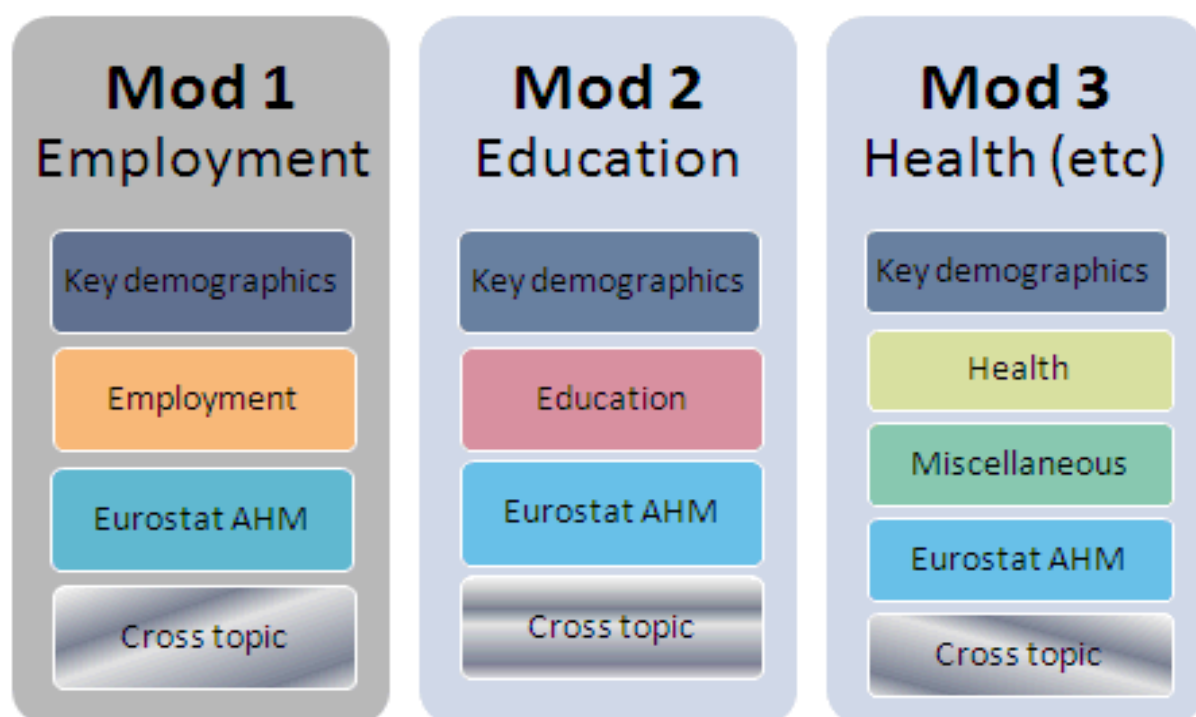
Modulising

The Labour Force Survey consists of approximately 600 variables which are potentially asked and approximately 300 derived variables. Currently, the average interview time for a wave 1 interview on the LFS is approximately 31 minutes for an individual and 75 minutes for household (30 minutes FtF and 34 minutes TO individual). Previous qualitative and quantitative ONS internet research concluded that 30 minutes is the maximum time respondents would be prepared to spend completing an internet questionnaire - for both individual and household questionnaires. The 2011 LFS internet pilot contained only the key ONS Labour Market Division questions and on average, respondents fully completed the survey in 18 minutes. This poses a problem for loading the full LFS questionnaire online when aiming to comply with the 30 minute recommendation.

As noted above, the LFS as it currently stands is too lengthy to load the full content into one questionnaire online. Modularising the LFS for an online mode entails creating more than one version of an online LFS questionnaire (to be issued in the same interview period to subgroups of the web sample). Each version of the online LFS is referred to as a 'module'. As depicted by Figure 1 below, each module would contain:

- A demographic core (e.g. age, sex, household grid, relationship grid etc)
- A cross topic set of questions, unique to the module, to enable key bivariate analysis (i.e. in the employment module this would comprise of variables required by LMD to run crosstabs on the employment variables)
- Coverage of a main theme, for example 'Employment' or 'Education' and light coverage of other topics, for example 'Illness' as opposed to exclusively covering topics. This would enable analysts to combine the main theme data from the module with the full topic data from the FtF and TO modes. This design also means that the full LFS questionnaire can be loaded online with no need to cut/reduce the number of topics covered in general, i.e. all topics will be asked but not to each respondent - hence modularised approach.

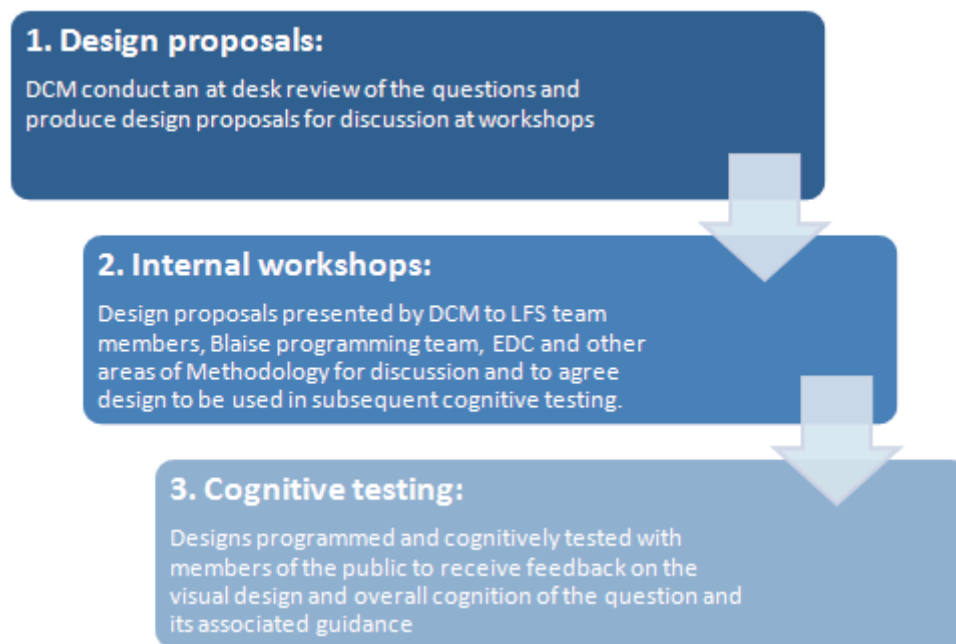
Figure 1. Modularised LFS:



Content for the Alpha pilot wave 1 questionnaire:

The employment module has been selected first for development. Part of this development work includes qualitative research whereby Data Collection Methodology will conduct a questionnaire review of the LFS questionnaire to ensure the questions are suitable for online completion. The questionnaire review comprises of three stages (see Figure 2):

Figure 2. Depiction of the questionnaire review process



As outlined above, the LFS contains a high number of variables and the questionnaire review process is lengthy and must be conducted in stages. The employment module will be used in the Alpha pilot (May 2015) however a full questionnaire review of the relevant questions will only have taken place for a limited number of variables. Figures 3 and 4 depict the recommended approach.

Figure 3. Type of review to be conducted

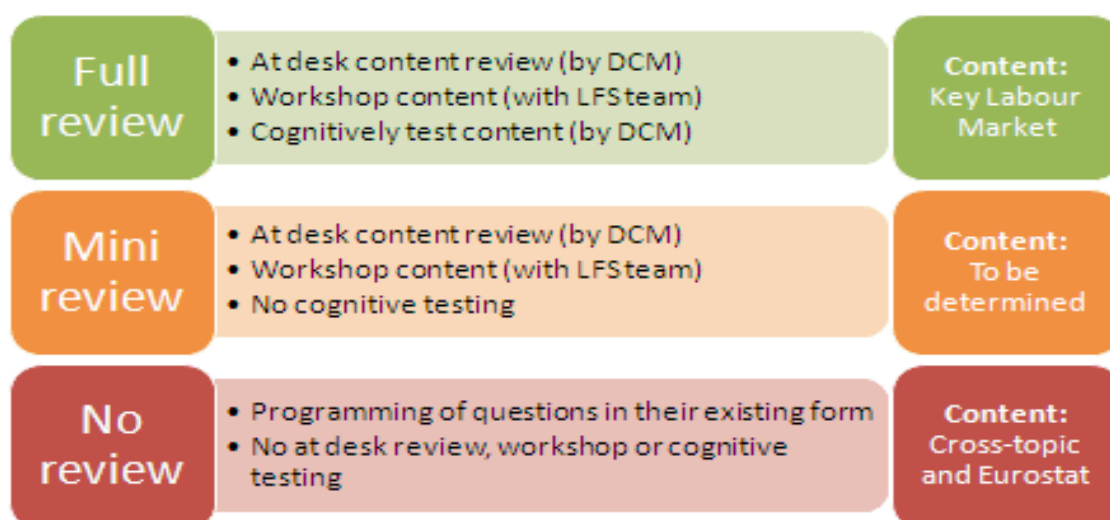
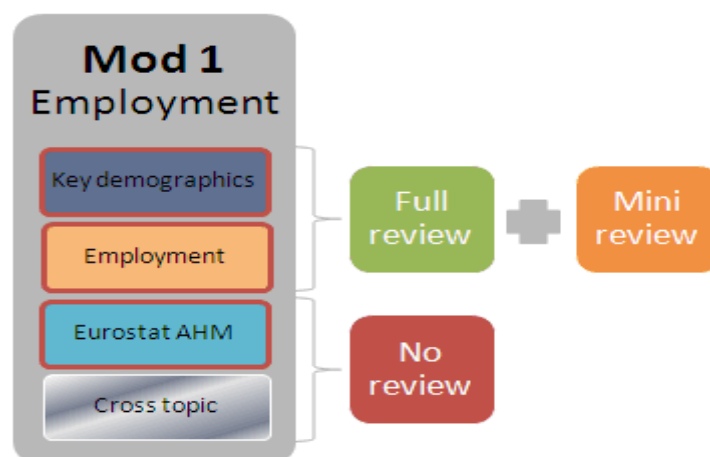


Figure 4. Questionnaire review approach for the Employment module (red border indicates content included in Alpha pilot):



A mixed review approach will be taken for the 'Key demographics' and 'Employment' sections of the questionnaire due to priorities. The 'Eurostat Ad Hoc Module' will not be reviewed for Alpha. The 'Cross topic' section will not be included in the Alpha pilot as the content of this section is yet to be determined; an extensive consultation exercise with clients is required to define the content.

Data Collection Methodology suppliers agreed that it was realistic to fully review approximately 20 variables in the run up to the pilot. Therefore Labour Market Division were consulted and asked to inform priorities and identify the key variables required for their analysis. Table 1 shows the list of prioritised variables.

Table 1. Variable overview

Variable name	Type of variable	Variable description
Typschr12	Question variable	Type of government training scheme the respondent is on
Schm12	Question variable	Asks whether the respondent was on a government training scheme for reference date
Sex	Question variable	Respondent gender
Age	Question variable	Respondent age
Yptjob	Question variable	Asks why they took a part time job rather than a full time job
Cured8	Derived variable	Current education received
Durun	Derived variable	Duration of ILO unemployment
Ftpt	Derived variable	Whether working full or part time
Inecac05	Derived variable	Economic activity (International definition)
Redund	Derived variable	Whether made redundant in last 3 months
Secjmb	Derived variable	Whether second job/status in second job
Sumhrs	Derived variable	Total actual hours worked in main and second job
Govtor	Derived variable	Government Office Regions

Although Labour Market Division was able to provide a greatly reduced list of variables, the majority of which are derived variables. After further investigation, to include all of the question variables that contribute to the derived variables would mean reviewing a total of 47 variables. It was not possible to review the full list ahead of the Alpha pilot. Labour Market Division were asked to reduce the list further and identified Inecac05 as the most important variable. An analysis was conducted to determine which variables contribute to Inecac05 and to identify potential overlap with the other variables listed. The result of this analysis is as follows, the blue highlights the duplication of the Inecac05 variables in the other key Labour Market Division variables:

Figure 5. Inecac05 composition analysis

Key Labour Market variables								
	Priority 3	Priority 2	Priority 2	Priority 1	Priority 3	Priority 2	Priority 3	N/A
DV name ▶	cured8	durun	ftpt	inecac05	redund	secjmbr	sumhrs	govtor
Feeder variables ▶	ioutcome	age	ioutcome	ioutcome	ioutcome	ioutcome	ioutcome	UALAD99 (from sample file)
▶	age	wrking	age	age	age	age	age	
▶	wrking	jbaway	wrking	wrking	refwky	wrking	jbaway	
▶	jbaway	ownbus	jbaway	jbaway	leftyr	jbaway	typsch12	
▶	ownbus	relbus	ownbus	ownbus	refwkm	typsch12	acthr2	
▶	relbus	country	relbus	relbus	leftm	ytetjb	totac1	
▶	course	schm12	typsch12	country	everwk	stat2	totac2	
▶	attend	fund12	ytetjb	schm12	redyl13	secjob	illwk	
▶	enroll	typsch12	ftptwk	fund12	redany	y2job	illdays1 - 7	
▶		ytetjb		typsch12	conmpy		everot	
▶		start		ytetjb	consey		actwkdy1 - 7	
▶		look4		start	conmon			
▶		lkyt4		look4	redpaid			
▶		lktimb		lkyt4				
▶		refwky		helpse				
▶		leftyr		statr				
▶		refwkm		noiwm				
▶		leftm		likewk				
▶		lktima		statr				
▶		everwk		ystart				
▶				wait				

Figure 5 illustrates that there is a great deal of duplication across the other variables identified by LMD. Therefore, DCM and EDC agreed the following approach for review:

Priority 1: INECAC05

Priority 2: DURUN

FTPT

SECJMBR

Priority 3: CURED8

REDUND (unlikely - variables not yet reviewed)

SUMHRS (" " ")

The variables for Inecac05 will be reviewed first, followed by the others listed in order of priority.

Cognitive Interviewing Approach

Principles

- Optimising is key - design to exploit / take advantage of each mode **not** unimode design
- You need to rethink how you redesign questionnaires
- Design for the future not the past (ignore 20, 30, 40 year old questionnaires)
- Been protected by the interviewers by asking poor questionnaires collecting poor data
- Need to reduce our expectations of respondents – this will help with non-response!

Recommended approach

- Embedded experiments – small, regular and often
- Quasi-experimental designs – before and after change analysis
- Don't forget to use your paradata (inform qual and quant work)
- Make use of labs for questionnaire testing, i.e. usability suites
- Conduct qualitative research – advocate mixing qual and quant for best insight and results

Much of our principles have drawn up through experience and in conjunction with an ESSnet (ESS - European Statistical System) Project on Data Collection for Social Surveys using Multi Modes. The ESSnet was initiated by Eurostat in the autumn of 2012 with a running time of two years and the project was shared by a consortium of Statistical Offices in five countries, Netherlands, United Kingdom, Norway, Finland and Germany. The cooperation between partners was very successful and discussion within the consortium helped improve instruments and methods. The final workshop was held on the 4th September 2014 and the papers can be found here:

<https://www.destatis.de/EN/AboutUs/Events/ESSnet/Agenda.html>

Based on the recommended approach of conducting embedded experiments which are small regular and often ONS has been conducting cognitive interviewing with small groups of respondents, up to 20), on an individual basis since 2011 with more to come. These sessions have focussed on:

- Visual design
- Colour and fonts
- Banners
- Help icons
- Question format / functionality
- Navigation
- Edit checks and validation

Procedure

The interviews consisted of observation of respondents' self-completion of the questionnaire on an ONS laptop (unassisted), interviewers making notes of respondent behaviour including how they dealt with various functions and question types, 'think aloud' comments made and difficulties encountered. This was followed by retrospective probing aided by a topic guide, covering: question comprehension and answer processes (including use of instructions and guidance and validation of answers); editing and validation checks; visual design and functionality; general reactions to the questionnaire and how it related to the respondent communications; and thoughts on being invited subsequently to take part in Wave 1 (hypothetical).

Analysis and Reporting

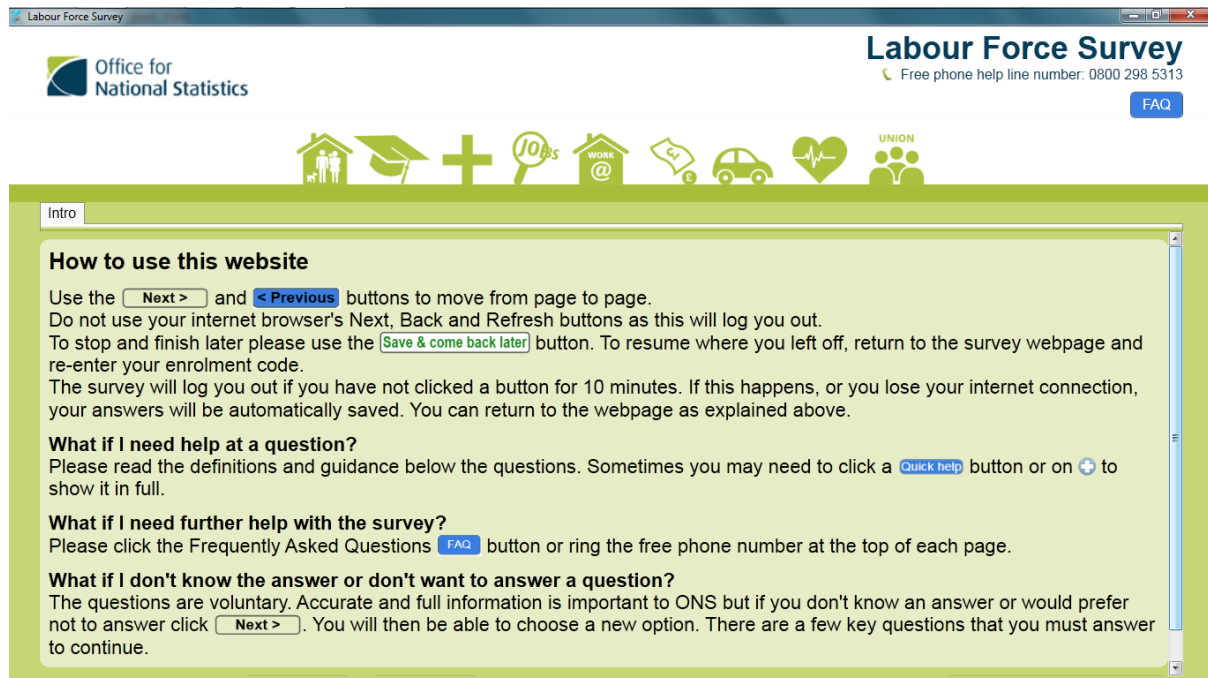
Interview recordings were then summarised into a framework chart and a thematic content analysis conducted.

For each question or information screen we present screenshots and then findings related to the topics covered together with points for discussion across the project development team. Where

possible recommendations for program changes are made. Overarching topics are reported at the end.

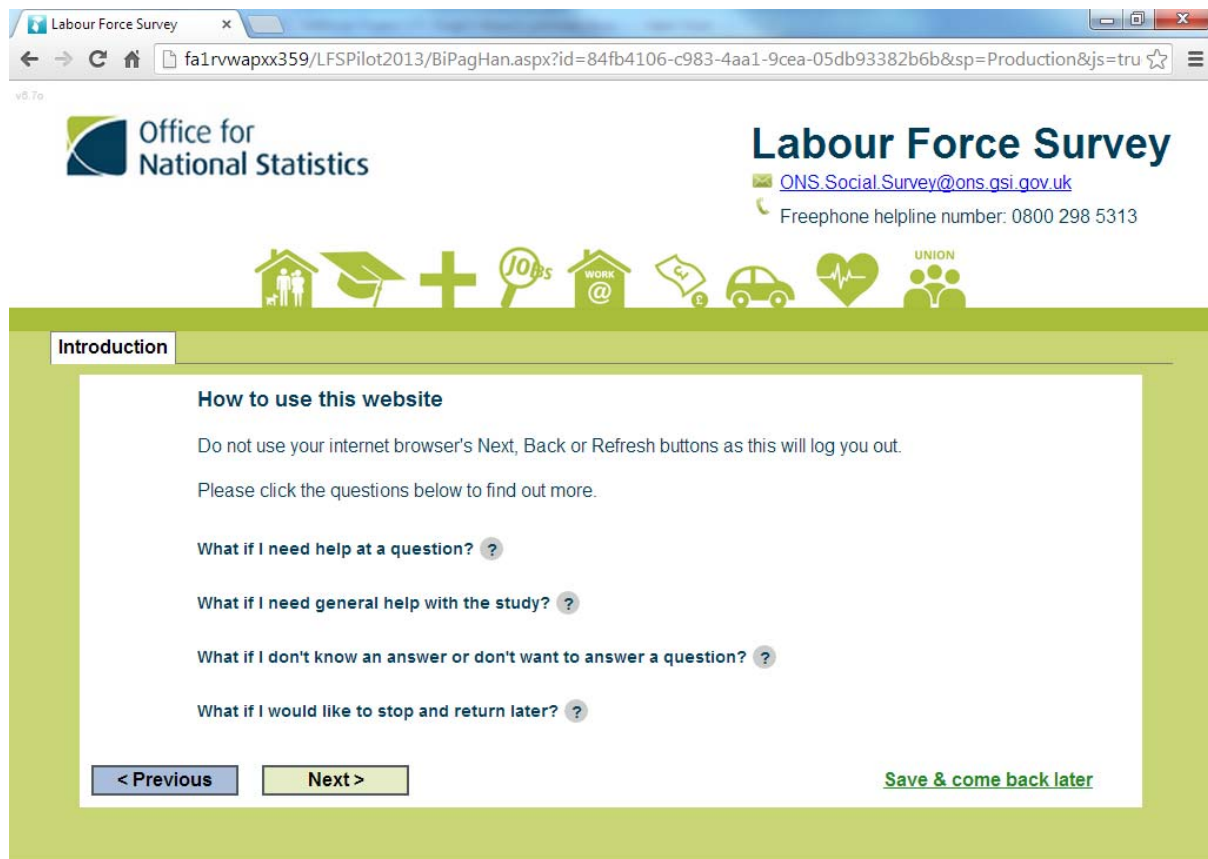
Below are some of the findings in relation to each area referred to previously.

Visual design



- Keep text to a minimum
- Grids with more than four respondents become cumbersome
- Placement of Help text needs to be thought through, more is not always helpful

Colour and fonts



- Respondents prefer a large font, but they did not struggle to read the questions. The exception to this was where there was a lot of information presented on screen, when respondents tended to lean forward and squint in order to read the text
- Ensure that fonts are consistent throughout, i.e. logo, banner and questions must all be the same
- Respondents liked the use of bold for the question to differentiate it from the answer categories and instructions
- The colour of the background is important it needs to be striking / eye catching / clear

Banners

- Utilise the screen space available for the question and answer text, respondents are frustrated by having to scroll down or read text that was too condensed
- It varied as to whether respondents noticed the 'FAQ' button
- Respondents saw FAQs as a preferable alternative to a helpline, and would use them before using a helpline
- More information on the helpline including opening hours was suggested
- Respondents liked the fact the FAQ and helpline remain present throughout

Help icons

- The presence of the 'i' icon for the instruction caused some confusion as the cursor changed to a hand when hovering over it, suggesting to respondents that they could click on it
- Respondents were not sure whether more information would become available to them were they to click on the button

Question format/functionality

- Respondents generally had no issues with the drop down selections but having radio buttons for questions where there were few options to choose from radio buttons were preferred as this is quicker than a drop down
- Ensure radio buttons are large enough to avoid respondents making keying errors
- Respondents found searching through long lists frustrating

Navigation

- Respondents want an indication of progress – we deliberately displayed none
- Suggestions included:
 - A percentage completion at the bottom of the screen.
 - A moving icon indicating the stage of questionnaire such as the car icon on the DVLA website.
 - An index on the side to show all the questions and enable respondents to go back to previous questions and indicate progress i.e. a progress 'tree' index at the side of the questionnaire.

Edit checks and validation messages

The screenshot shows a web browser window titled 'Labour Force Survey' with the URL 'fa1rvwapxx359/LFSPilot2013/BiPagHan.aspx?id=84fb4106-c983-4aa1-9cea-05db93382b6b&sp=Production&js=tru'. The page header includes the 'Office for National Statistics' logo and the 'Labour Force Survey' title with contact information: 'ONS.Social.Survey@ons.gsi.gov.uk' and 'Freephone helpline number: 0800 298 5313'. Below the header is a row of icons representing different life stages: a family, a graduation cap, a plus sign, a magnifying glass with '10yrs', a house with 'WORK @', a car, a heart with a pulse line, and a group of people labeled 'UNION'. The main content area has two tabs: 'Introduction' and 'Household details'. The 'Household details' tab is active, showing a form with the following fields: 'First Name' (containing 'M'), 'Surname' (containing 'H'), 'Male/Female' (with 'Male' selected), and 'Date of birth' (with separate boxes for 'Date', 'Month', and 'Year (YYYY)'). The 'Date' box contains '4/5' and has a red error message: 'Please enter a valid date.' The 'Month' box is a dropdown menu with 'Select' and has a red error message: 'Please answer before continuing.' The 'Year' box contains 'YYYY' and has a red error message: 'Please answer before continuing.' At the bottom of the form are three buttons: '< Previous', 'Next >', and 'Save & come back later'.

- Respondents considered the design of the error message in terms of colour was effective.

- Make error messages eye catching.
- Ensure the wordings of messages are clear and straightforward
- Respondents preferred having the error messages appear when completing the question rather than at the end of the
- Error messages in grid are confusing to respondents, they cause confusion and scrolling
- Provide explicit information / guidance on what buttons will do
- Respondents require a clear 'confirmation of completion' page

ESSnet conclusions

A web Labour Force Survey is possible, but there are obstacles

Mixed mode is the future – not web only due to bias and poor response rates

Web achieves the same demographics as current modes

There are mode effects but we shouldn't allow them to stop us

Time series **will** be compromised and trends breached, this will be continuous, we need NSI's and Eurostat to help customers understand and to communicate this.

The ESSnet DCSS project has come to an end and many of the countries in the ESS plan to introduce web as part of their mixed-mode designs for social surveys, and based on the findings of the project there seems to be no fundamental reason to advise against this strategy.

In 2014, many of the important challenges identified at the beginning of the project still remain. These include web questionnaire design issues, such as, finding effective uses of instructions to explain complex statistical concepts to subgroups among web respondents or devising optimal navigation for complex household questionnaires. Also many IT-related challenges continue to exist or have become even more relevant, such as finding a strategy to deal with respondents wishing to complete web questionnaires on mobile devices or the development of good software solutions for case management in mixed-mode designs.

Working with Blaise 5

The Office for National Statistics (ONS) has used two other packages to test collecting data on the web, Blaise IS (the internet mode for Blaise 4) and ConfirmIT. Both have shown that it can be done, and screen designs with certain constraints can be achieved. Blaise 5 is a step change in capability when compared to Blaise IS, but it is not without 'features' which some might define as bugs. Given the screen designs we were aiming to achieve (visual appearance being key to respondent take up), and given what we knew about Blaise IS and ConfirmIT and with the software we had available, we decided that we would attempt the Labour Force Survey with Blaise 5.

ONS has an excellent working relationship with the Blaise development team at Statistics Netherlands and I believe both organisations have benefited from the development of the Labour Force Survey in Blaise 5. In some respects we have acted as a testing team for the developers spotting bugs / defects and feeding back the results to them as new releases of Blaise 5 have become available. ONS has benefited by highlighting that certain functionality was missing and this has been included in future releases.

If I had a criticism of the development process it would be that some versions of Blaise 5 have been released with insufficient internal testing by the developers. This has led to a loss of confidence in

the product internally within ONS and given that Blaise 5 lacks a multi-mode capability, something competing products have, is causing questions to be asked about the future of Blaise within ONS.

This is also a question for other NSI's and I was pleased to hear that Statistics Netherlands has committed to Blaise 5 as a strategic solution for their data collection and an increase in the size of the development team. I look forward to hearing how the Blaise team will tackle multi mode case management within Blaise 5 as multi mode is the future.

It is not all gloom and doom however and ONS will be running our first Blaise 5 Pilot in June, our questionnaire has been signed off by the customer and feedback we have received from ESSnet and testing indicates that we have a product that looks modern, inviting and has some clever features.

Bibliography

Wilson, Phelps: ESSnet workshop – Data Collection for Social Surveys using Multiple Modes 4-5
September 2014, Wiesbaden, Germany

Wilson, Simmonds, Barford – EDC LFS Wave 1 eQ Development approach and content

Data Collection Methodology - LFS Online Testing Stage 1 (Visual Design and Functionality) – Wave 2

Blanke, Luiten – Essnet workshop – ESSnet DCSS: Lessons so far

Blaise 5 Server Configuration for Web Surveys

*Paul Segel, Mangal Subramanian, Richard Frey, Ray Snowden
Westat*

A configuration of Blaise 5 software used to support web surveys and connected mobile surveys is referred to as a Blaise Server Park. Blaise uses a service oriented architecture and the various major functional components of the Blaise software are programmed, installed, and operated as services to permit the software to be installed on one or several servers. This flexibility permits organizations to adopt various server configurations of Blaise 5 to meet different scaling, continuity, and security requirements.

This document is written primarily with reference to the installation of Blaise 5 using Microsoft platforms for web surveys.

A Blaise 5 Server Park consists of one or several Windows servers on which the following Blaise 5 server roles are configured:

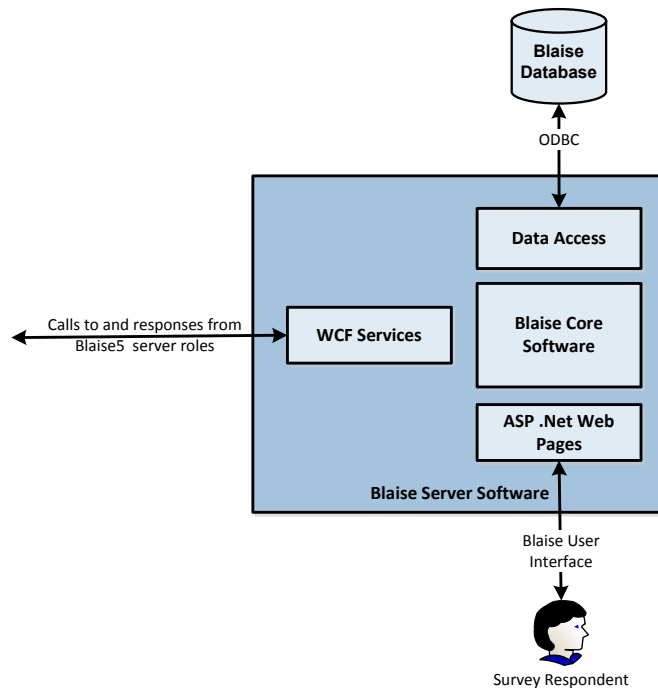
Blaise Server Role	Description
Survey Manager	The Survey Manager role provides administrative functions to manage the Blaise 5 server park including defining server roles, maintaining users, creating and publishing surveys, etc.
Web server	The Web server role hosts files for Blaise internet surveys. This is the server role to which end-users directly connect. The Web server role processes user input and sends UI responses.
Resource server	The Resource server role applies layout and formats Blaise 5 user interface output for different devices/modes.
Data Entry server	The Data Entry server role interprets and validates input and applies the rules defined in the survey.
Data server	The Data server role reads data from and writes survey data to a Blaise 5 database.
Session Server	The Session server role manages and stores data associated with all active interview sessions. In most cases, the Session server role should be assigned to the same Windows server that the Data server role has been assigned.
Audit Trail Server	The Audit Trail server role maintains a log of audit records, each of which contains evidence directly pertaining to and resulting from the execution of a Data Entry Session.
Database Management System (DBMS)	Although not part of the Blaise 5 Server Park, a DBMS is also required to host the Blaise database. Blaise 5 currently supports SQL Lite (installs with Blaise 5 by default), Microsoft SQL Server, MySQL, Sybase, and Oracle.

It is important to note that the Blaise 5 server roles are functional designations of the Blaise software and do not correspond to the underlying Windows servers. A Blaise 5 Server Park can consist of one or several Windows servers depending on the level of resource required and the configuration and security standards of the organization. The following are additional options and restrictions for the configuration of Blaise Server Parks:

- All Blaise 5 server roles can be installed on a single server and a DBMS can also be installed on the same server. As a result, a fully functioning Blaise 5 server installation can be supported on a single server.
- For improved security, it is common practice to isolate DBMS servers from front-end web servers through the use of firewalls or other network devices. As a result, a minimum recommended configuration of Blaise 5 for production use would consist of at least two servers – one server for the Blaise 5 software and a DBMS server.
- Alternatively, Blaise 5 server roles can be installed on separate servers. Web, Resource and Data Entry server roles can also be installed on multiple servers for enhanced scalability and continuity.
- The Data server role, Session server role, Audit trail server role, and Survey Manager role can each be installed on only one server in a Server Park.
- One server in a Blaise 5 Server Park is established as the Survey Manager for the Server Park. Functions in the Survey Manager are used to configure other servers on which Blaise 5 software has been installed, i.e. which servers will support which server roles and the protocols and ports to be used.
- Scaling/clustering of the DBMS server is a function of the DBMS software and not directly supported by the Blaise 5 software. Scaling/clustering of the DBMS server must be configured and supported by a Database Administrator familiar with the capabilities of the platform being used.
- Blaise 5 server roles can be installed on either physical or virtual servers depending on the capabilities of the underlying operating system being used. Similarly, Blaise 5 server roles can be installed on Windows servers hosted using cloud IaaS services such as Amazon AWS or Microsoft Azure. The use of cloud services may present unique security, connectivity, cost, and operations support issues that must be considered.

Blaise Server Software Architecture

The following diagram shows the major functional components of the Blaise 5 server software and the mechanisms used by the software to communicate with/connect to external parties/resources:



- The Blaise 5 installation and software footprint is the same for every server that will be used to support a server role in a Blaise server park. In order to prepare a server to function as part of a Blaise 5 Server Park, a full installation of the Blaise 5 server software is performed. Blaise 5 server roles are defined on the server during the installation process.
- A Windows service is created when the Blaise 5 software is installed. It is through interfaces exposed by this Windows service that Blaise 5 server role functionality is invoked through WCF service calls.
- Blaise 5 is a .Net 4.0 application and the web functionality is supported via an ASP .Net web framework running under IIS 6.0 or higher.
- The DBMS software and related OS components are not shown in this diagram but are necessary and central to the operation of the software, i.e., for a Windows installation, a Blaise Windows service is created during software installation and supports the various service interfaces, Internet Information Services (IIS) accepts connections from the end user and invokes the appropriate Blaise 5 ASP .Net pages, and SQL Server provides DBMS services to the Blaise database.

Connectivity requirements in a Blaise Server Park

The following table indicates the connectivity requirements between the various server roles in a Blaise 5 Server Park and with key external resources. The information in this table is particularly useful when planning for the installation of Blaise 5 server roles on multiple servers. Where a single Blaise 5 application server is used (i.e., all server roles running on a single server) most of these interactions are made internally on the server.

Calling Role Service Role	Web	Resource	Data Entry	Data	Server Manager	Session	Audit Trail	DBMS	Web User
Web	--	--	--	--	HTTP/ 8031	--	--	--	HTTP/ HTTPS 80/443
Resource	HTTP/TCP 8033	--	--	--	HTTP/TCP 8031	--	--	--	--
Data Entry	HTTP/TCP 8033	--	--	--	HTTP/TCP 8031	--	--	--	--
Data	--	--	HTTP/TCP 8032	--	HTTP/TCP 8031	--	--	--	--
Server Manager	--	--	--	--	--	--	--	--	--
Session	--	--	HTTP/TCP 8032	--	HTTP/TCP 8031	--	--	--	--
Audit Trail	HTTP/TCP 8033	--	--	--	HTTP/TCP 8031	--	--	--	--
DBMS	--	--	--	ODBC	--	ODBC	ODBC	--	--
Web User	--	--	--	--	--	--	--	--	--

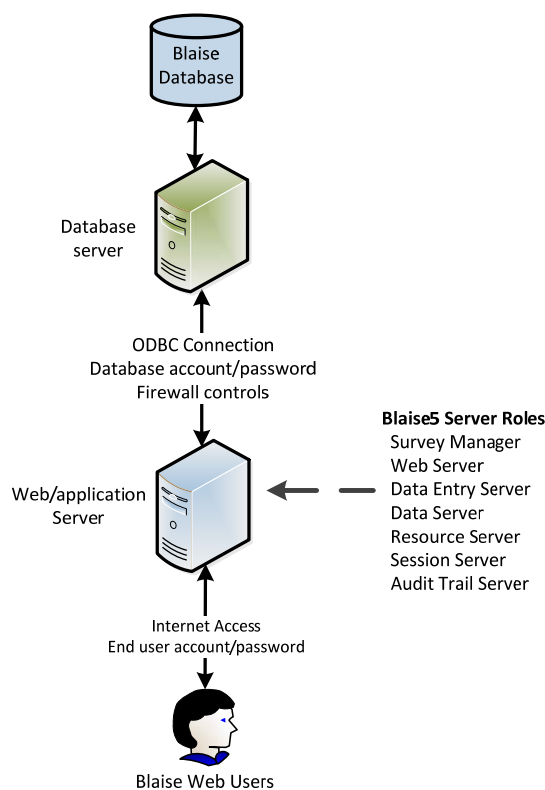
Additional notes on connectivity requirements between Blaise 5 server roles:

- The server roles/resource listed in the column is the resource initiating the call; the server role/resource listed in the row is the resource servicing the call.
- A cell value of '--' indicates that the calling resource (column) will never request a connection to the serving resource (row). For example, resources never call themselves (all cells on the diagonal have the value '--'), the Web server role does request service from the Data Entry server role, the Resource server role does request service from any server role, etc.
- Each cell in the table lists the protocol (or protocol alternative) and the default port over which the connection is made. When configuring the server roles on servers you can designate the protocol and port to use and override the default value.
- Only two server roles support connections directly with end users –the Web server role for web survey respondents over HTTP/HTTPS and the Survey Manager through a Windows application installed on the server.

Blaise 5 Server Configurations

The Blaise 5 architecture is quite flexible and permits organizations to adopt several different server configurations based on resource availability, scaling considerations, security requirements, or other factors.

2-server configuration - the following diagram shows a 2-server configuration for Blaise 5. One server is a web/application server on which the Blaise 5 software is installed and which support all of the Blaise 5 server roles. The 2nd server is a database server running SQL Server or another supported DBMS:

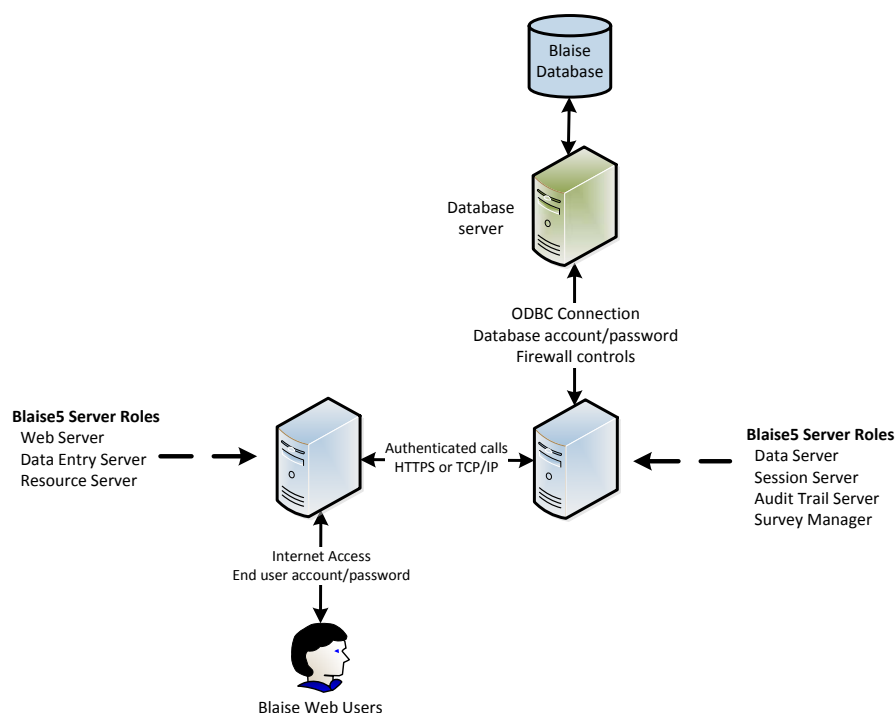


Notes about this configuration:

- This represents the minimum recommended security configuration for production use as it permits the database to be segregated from the front-end web server.
- A single instance of SQL Server can be used to support multiple Blaise 5 surveys and a separate SQL Server database can be defined for each Blaise 5 survey. This permits organizations running Blaise 5 to host several different surveys on the same underlying DBMS server and control access to survey data through the creation of separate SQL accounts and access control rules.

- During installation, the Blaise 5 software is configured to run under a Windows account. By default the local administrator account is used. Alternatively, a new Windows account can be created and assigned only the privileges required to run the Blaise 5 software. This is consistent with the security principle of least privilege and is a more secure practice.
- The database connection between the Blaise 5 Data server and SQL Server is secured through the use of an account/password which is stored in a Blaise .bdix file.
- Access to the Blaise 5 instrument by Blaise users can be configured to use an HTTPS connection so that all user input is encrypted.
- The Survey Manager provides functions to create end-user accounts with passwords. These accounts can be assigned to end-users to control access to surveys.
- Stress testing performed by Westat indicates that a 2-server configuration can support between 200 – 300 concurrent users with acceptable response times. These results depend on the complexity and coding of the web survey, specification of the servers, etc.

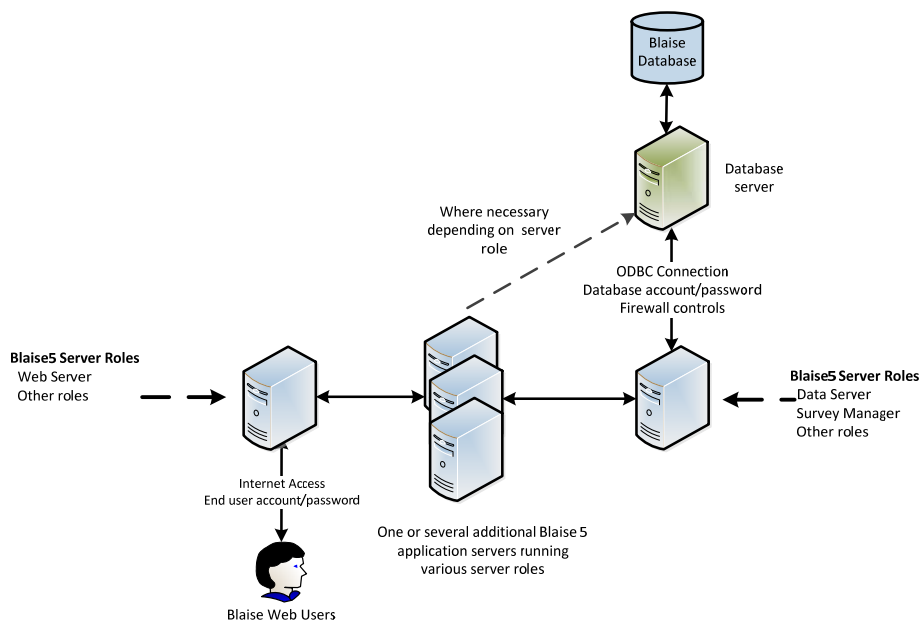
3-server configuration - the following diagram shows a 3-server configuration for Blaise 5 (one database server and two Blaise 5 web/application servers) with a recommended allocation of Blaise 5 server roles between servers:



Notes about this configuration:

- This configuration provides additional server resources for greater scale.
- This configuration also segregates the Blaise 5 server roles that access the Blaise 5 database from direct end-user access.
- Interaction between the Blaise 5 server roles on the two Blaise 5 servers is over HTTPS or TCP/IP protocol and can be secured through account/password or certificates.
- Westat has not yet performed stress testing using a 3-server configuration. We plan to complete this over the next several months.

Configurations for greater scalability – the following diagram shows a configuration where additional Blaise 5 servers have been added to the Server Park to scale out beyond a 3-server configuration:



Notes about this configuration:

- The Survey Manager role and Data Server role, Session Server role and the Audit Trail Server role can each be assigned to only one server in a server park.
- The Data Entry server role and Session server role are the two server roles that tend to become resource intensive as the number of concurrent users is increased. In instances where it becomes necessary to add additional servers to a Blaise 5 server park (i.e.,

more than 3) it will typically be most beneficial to assign the Session server role to a dedicated server and the Data Entry server role to multiple servers.

- Where the Web server role is assigned to more than one server, this configuration is analogous to employing the use of a network load balancer to distribute user connections to a web site across multiple web servers. The Blaise 5 software provides the functions of balancing load and distributing sessions across the multiple server.

Westat is continuing to work with Blaise 5 software under different configurations to better understand the various configuration options and more completely explore the performance of Blaise 5 for web surveys under increased load.

Tourist Survey from Blaise 4.8 to Blaise 5 in Norway.

Jan Haslund, Statistics Norway

Introduction of the tourist survey in Norway

In the tourist survey in Norway we collect data from tourists visiting Norway. The tourists have to overnight at hotel or camping. Our sample is hotels and camping sites in Norway. The sample is about 200 sites each four months. The interviewers travel to the hotels and in the morning (at breakfast) they ask the visitors if they will participate in the survey. The visitors then answer the survey using tablets or paper. The reason for using pen and paper is that we do not have enough tablets. Some places there are also bad network connection and mobile connection. The respondents are both tourists and people on a business trip.

The Blaise 4.8 solution

At the Blaise Conference in London 2012 we saw a presentation of C-Moto from CentERdata we decided to use this. So until 2015 our electronic questionnaire was an internet version using C-MOTO and Blaise 4.8. The interviewers also had the opportunity to give the tourist a card with the link to the questionnaire with UserID and password. The link was also on a QR code so they could read it with their smartphone. This version of the questionnaire has a little change in the primary key. (The respondent has to select the hotel themselves). The Primary key is hotel_ID, date and time. The questionnaire is in 14 languages. We also had two versions one for our own tablets and one for the tourists using the QR code. In addition we have a version for data entry of the paper forms. This means we had 29 different questionnaires in 14 different languages and in 29 different Blaise databases. We choose to have it in different databases because then we could start with one language and put them in production one by one. We could also easily change a questionnaire in one language if it was something wrong with it. Every morning we run some manipula scripts. The manipula scripts make one database of the 29 databases.

Because it was so many questionnaires which could change every fourth month we think it would be nice to generate them. For the generation we made a template for the Blaise questionnaire with .bla and .inc files. We have two .bla files one for or tablets and one for them who use their own smartphones or tablets. In the files we put tags like (" ##T08bTxt##") instead of texts. (Fig 1)

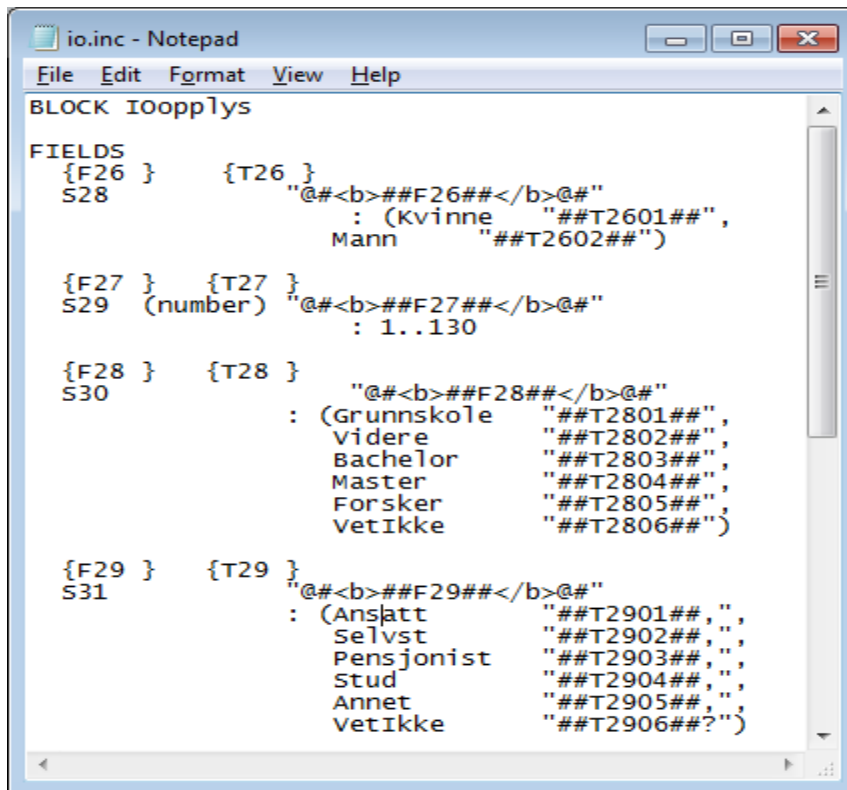


Fig 1

We put ## as a start signal for the tag and also as a stop signal. Doing it like this we will know that the text between ## and ## is the tag. These tags we would like to change for each language. We made an excel sheet with the tags and texts in all the languages (Fig 2).

	E	F	G	H	I	L	N
1			Norsk	Engelsk	Fransk	Koreansk	Kinesisk
399	T2902		Selvstendig næringsdrivende	Self-employed	Travailleur indépendant	자영업	自营
400	T2903		Pensjonist	Retired	Retraité	정년퇴직자	退休人员
401	T2904		Student/elev	Student/pupil	Etudiant/élève	학생	学生
402	T2905		Annet	Other	Autre	다른 직종	其它
403	T2906		Vet ikke	Don't know	Ne sais pas	모름	不明
404	F26		Er du kvinne eller mann?	Are you a woman or a man?	Êtes-vous une femme ou un homme?	본인은 여성 아니면 남성이세요?	您性别为?
405	F27		Hvor gammel er du?	How old are you?	Quel âge avez-vous?	본인의 연령?	您几岁?

Fig 2

The idea is that, if we should change the text in our questionnaire. We can change the text in Excel in one language or in all languages and generate new questionnaires we need. We will then have the same structure for the database for all languages since we use the same template for every language.

To generate the Blaise source files we made an AUTOIT script. This script read the Excel sheet into an array. It reads the template into an array (line by line). It looped thru the array looking for tags, when it find a tag it search thru the array from the excel sheet and find the text for the tag in the chosen language (This language is sent as a parameter). Then it swaps the tag with the text from Excel. When it has looped thru the whole file and changed everything it writes a file in a new directory. The directory has a name which identifies the language. Then it starts with the next template. We have one bla file (in two versions) and four .inc files. When all files are changed the .bla file is prepared. The AUTOIT script also prepares a manipula script. This script makes a new empty database. We need this empty database for the internet workshop. Then we have to do some small changes in the internet workshop and create a package. The package is then ready to be installed on our server. In the script you can select to convert one language or all languages at the same time.

Here is a short description of AUTOIT.

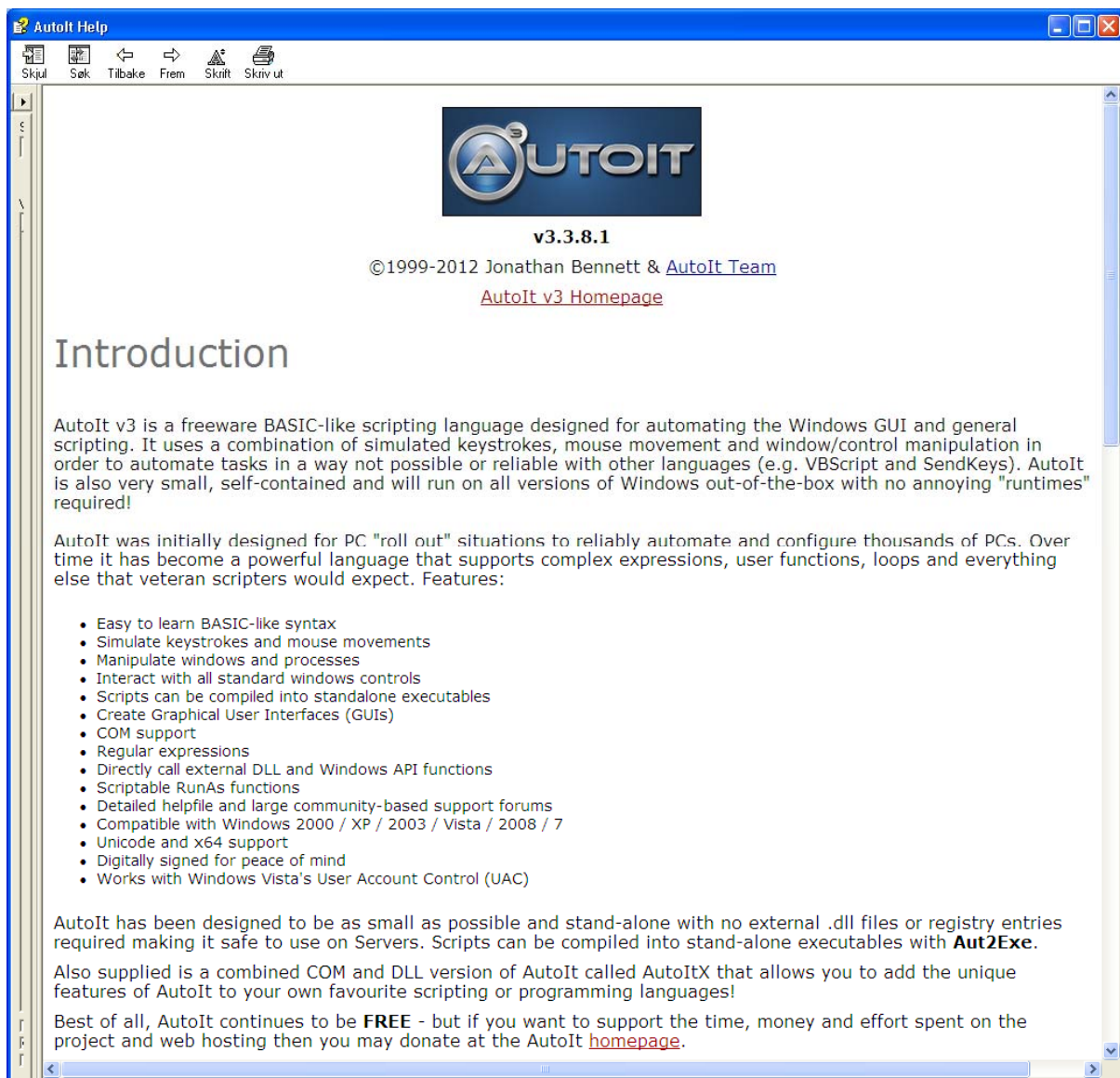
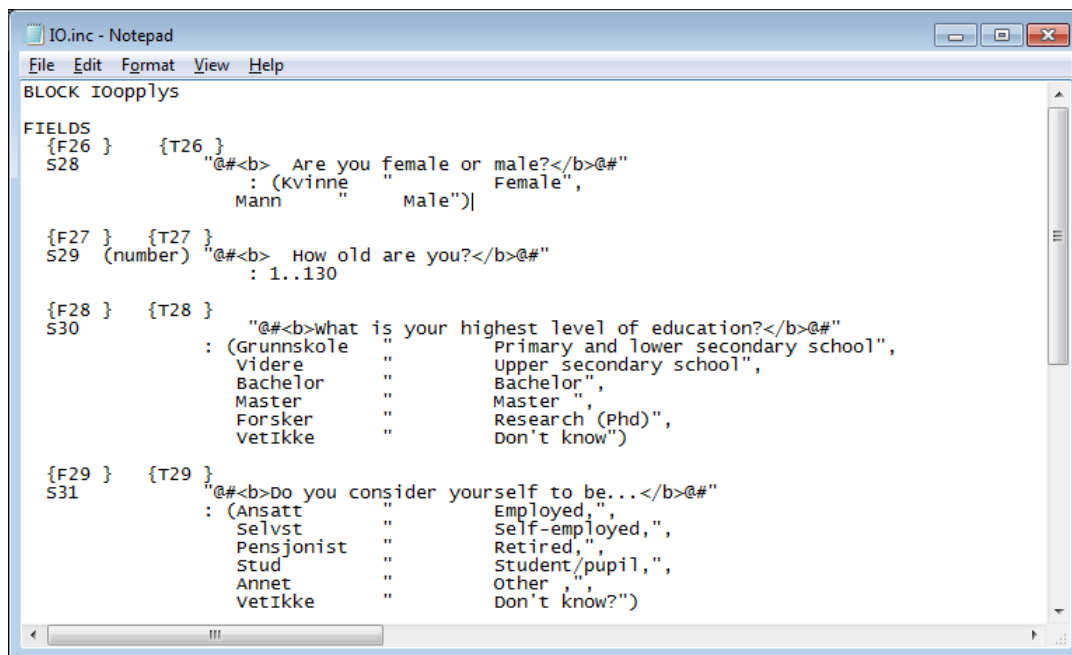


Fig 3

After we have ran the AUTOIT Script the file would look like this in English



```

IO.inc - Notepad
File Edit Format View Help
BLOCK Ioopplys

FIELDS
{F26 } {T26 }
S28      "@#<b> Are you female or male?</b>@#"
          : (kvinne "Female",
            Mann    "Male")

{F27 } {T27 }
S29 (number) "@#<b> How old are you?</b>@#"
          : 1..130

{F28 } {T28 }
S30      "@#<b>what is your highest level of education?</b>@#"
          : (Grunnskole "Primary and lower secondary school",
            Videre      "Upper secondary school",
            Bachelor    "Bachelor",
            Master       "Master",
            Forsker      "Research (Phd)",
            vetIkke      "Don't know")

{F29 } {T29 }
S31      "@#<b>Do you consider yourself to be...</b>@#"
          : (Ansatt      "Employed",
            Selvst       "Self-employed",
            Pensjonist    "Retired",
            Stud          "Student/pupil",
            Annet         "Other",
            vetIkke       "Don't know?")
  
```

Fig 4

Or like this in Chinese.



```

IO.inc - Notepad
File Edit Format View Help
BLOCK Ioopplys

FIELDS
{F26 } {T26 }
S28      "@#<b> 您的性别?</b>@#"
          : (kvinne "女",
            Mann    "男")

{F27 } {T27 }
S29 (number) "@#<b> 您的年龄?</b>@#"
          : 1..130

{F28 } {T28 }
S30      "@#<b> 您的最高学历?</b>@#"
          : (Grunnskole "初中",
            Videre      "高中",
            Bachelor    "学士",
            Master       "硕士",
            Forsker      "博士/研究生",
            vetIkke      "不确定")

{F29 } {T29 }
S31      "@#<b> 您的职业?</b>@#"
          : (Ansatt      "工薪阶层",
            Selvst       "个体工商户",
            Pensjonist    "退休人员",
            Stud          "学生",
            Annet         "其它",
            vetIkke       "不确定?")
  
```

Fig 5

For starting the questionnaire we have two asp files .In the first asp file, the interviewer selects the hotel where the interviewing is taking place. This is only set the first time the interview is started. The next asp is the page where the respondents start. On this page the respondent could click a flag for the language he or she want to use. After that there was not possible to change language.



Fig 6

Here is a screenshot from the survey in Blaise 4.8

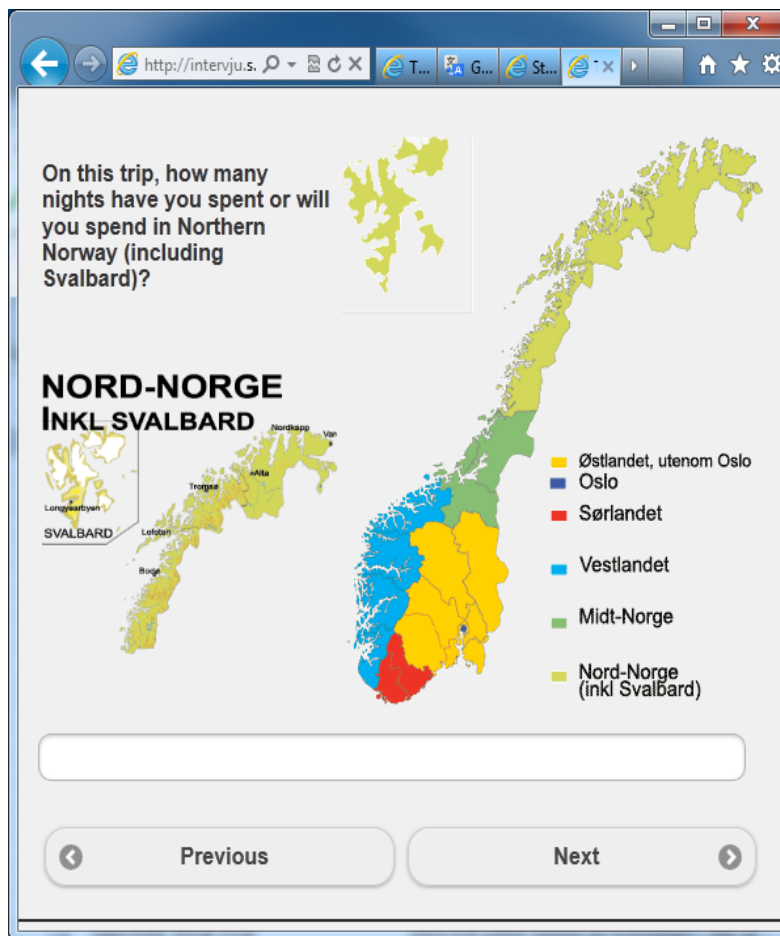


Fig 7

Converting to Blaise 5

When we should change the questionnaire for 2015 we decided that we will try to make it in Blaise 5. Instead of making it for browser, like the old version, we try to make it as an android app in Blaise 5. When we use an app we don't need a connection to the internet and could do the interviewing off-line and synchronize it at home. This means that we could use less pen and paper and more tablets. With everything in an app, we also would like to have all languages in one app and one database.

The first thing we will do was to make a version with all languages in one datamodel. We also want to generate it at least for the first time. We then took the AUTOIT Script and made some small changes to it. Instead of changing the tag with the text for one language we changed it with all languages (fig 8).

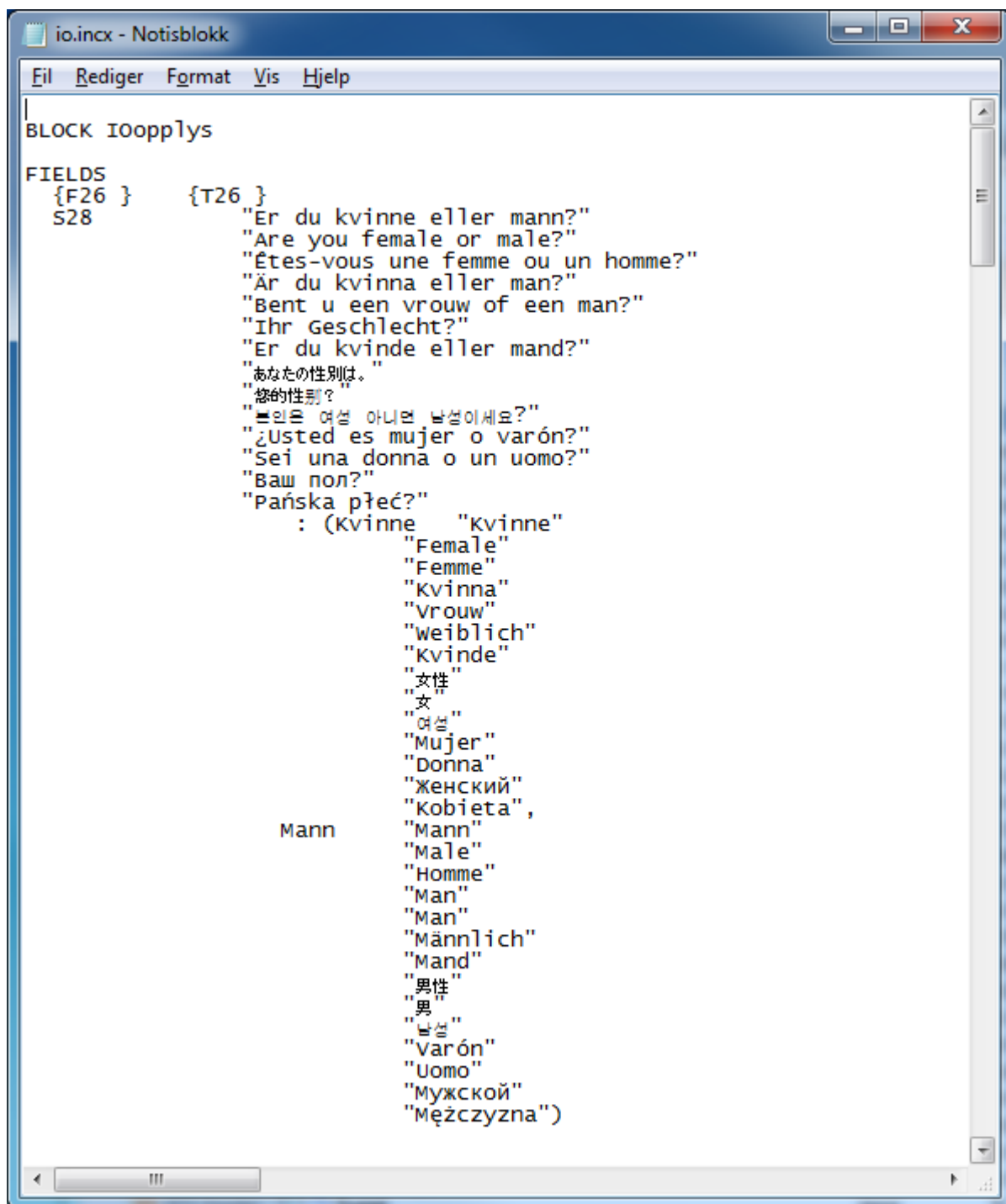


Fig 8

After we had generated it in Blaise 4.8 we converted it to Blaise 5 using Blaise4To5Source.exe. The conversion worked pretty well. When we looked at it in the layout section in Blaise 5 we find out that it was some problems with some html tables with two pictures and some texts (Fig 8).



Fig 9

To solve this we made a new gif with the two pictures and put the question text below the pictures.



Fig 10

Since the questionnaire already was used on a tablet it was quite simple, with mostly one question on each page. In addition it was some small tables. So it was not a big job with the Blaise code. We added some groups and start working with the layout. The first thing we did was making a new masterpage. On this page we wanted that the respondent could change language for the questionnaire. So we make a ribbon with flags, one flag for each language. The flags were image controls. We put an “OnClick” event on the control which changed the language to the language correspondent with the flag. Doing it like this the respondent could change language all time thru the interview.



Fig 11

We also have to put the 14 languages in the Blaise Resource database and translate some texts like next, previous etc. These texts will also change with the language the respondent select. We also made some small changes in the Blaise Resource database with colors, font sizes etc.

To start the survey in Blaise 4.8 we use two asp files. In Blaise 5 we could use the Intro page and Receipt page. The Intro page is Page with a text and a button. The text just says “Press start to start a new questionnaire”. The button says “start new questionnaire”. We use this start page because we want to have a start point for the respondents. The first time the survey is started we start the questionnaire with the primary key. The primary key is like in Blaise 4.8 the hotel_id, date and time. The interviewer has to select hotel and the date and time will be set to sysdate and systime. Then the first respondent can fill in the questionnaire. At the end the respondent came to the receipt page. This Page has an “OnLoad” event that starts a new survey with parameter for Hotel_ID, date and time. And it will go to the intro page for the survey. Since the field in primary key already is set we now start directly on the first question in the questionnaire.

We have now made a questionnaire and made some layout that we think looked ok on an android tablet, but we haven’t seen it on a tablet as an app yet. We have seen it in a browser from the tablet but we know it would not look the same in an app. In December 2014 we get an android client that we could try. We downloaded the client and installed it on our tablet and put the survey on our server. At first we run it on local network at the office. This way it was easy to make changes when something came up in the testing. We also have some problems with our server and network. First we have to open some ports. Still we have some problems connecting the tablets from the network to the server. It works if we used mobile data or connected to another network for example at home. So this is a problem with our network. The interviewers should not be on this network so it was not a problem after all. Here are some screen shots from our new Blaise 5 survey.



Fig 12

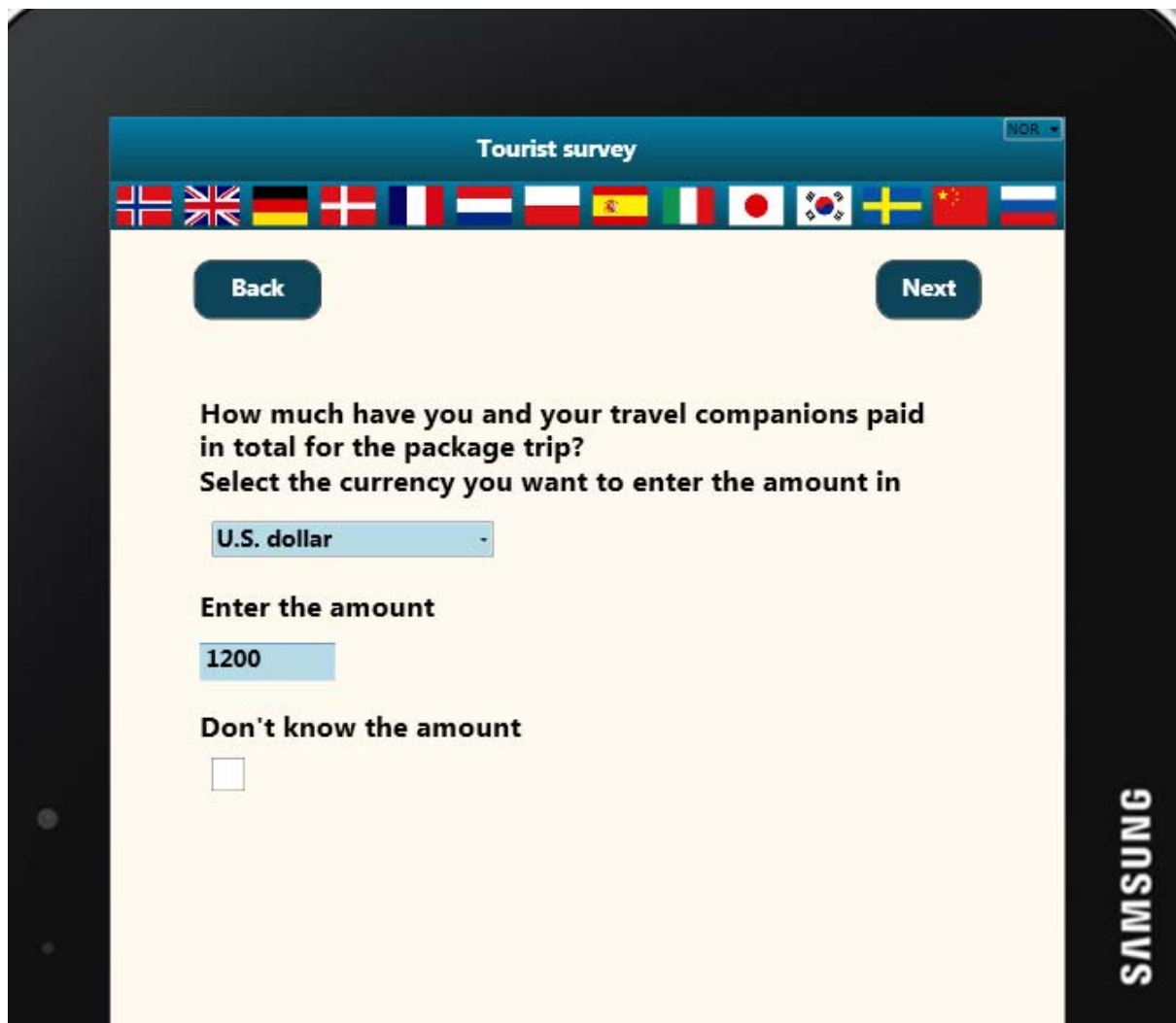


Fig 13

Reference

Alerk Amin, Arnaud Wijnant (CenERdata, Tilburg, The Netherlands) "Blaise On-the-Go Using Blaise IS With Mobile Devices", Proceedings of the 14th International Blaise Users Conference, April 2012

Rendering a Blaise 5 Survey on Different Devices

Richard Frey, Kathleen O'Reagan and Karen Robbins, Westat

Summary

Blaise is a powerful and flexible system used for computer-assisted survey processing in various modes like Computer-Assisted Personal Interviewing (CAPI), Computer-Assisted Self-Interviewing (CASI), and Computer-Assisted Web-Interviewing (CAWI) on various devices like laptops, tablets or smart phones. Rendering surveys on these devices requires survey management planning, designing for the general features of the target devices and implementing the resources.

The focus of this paper is to describe the basic modifications that need to be made for devices such as a smart phone or tablet. In addition we'll point out the pitfalls inherent to the device and ways to get around them such as dealing with tables and grids on a phone, scrolling and the use of arrows versus next/previous buttons. The paper will also discuss considerations when running a Blaise survey from a native Blaise App, either in connected or disconnected mode.

Introduction

With the introduction of Blaise 5, Blaise has moved from the constrained development and deployment environment of Blaise 4 to an expanded user friendly and accessible environment of mobile devices and tablets. The new visual elements of Blaise 5 have made it possible to present surveys on these devices, as well as enhance those that run on browsers and desktops.

This advancement of survey presentation is made possible with a more integrated approach to survey layout, significantly improved presentation customization capabilities and better support for mixed-mode surveys.

However, the widespread adoption of mobile devices, such as smartphones and tablets, is having a major impact on how survey research is done. Because these devices are lightweight and relatively inexpensive, they make it easy to collect data using such existing survey modes as computer-assisted personal interviewing (CAPI).

There are also a number of challenges associated with utilizing mobile devices for survey data collection:

Can respondents effectively use them? The input methods that these devices permit, particularly their touchscreen interfaces, may be relatively unfamiliar to some users (e.g. older respondents) and may create difficulties interacting with the device

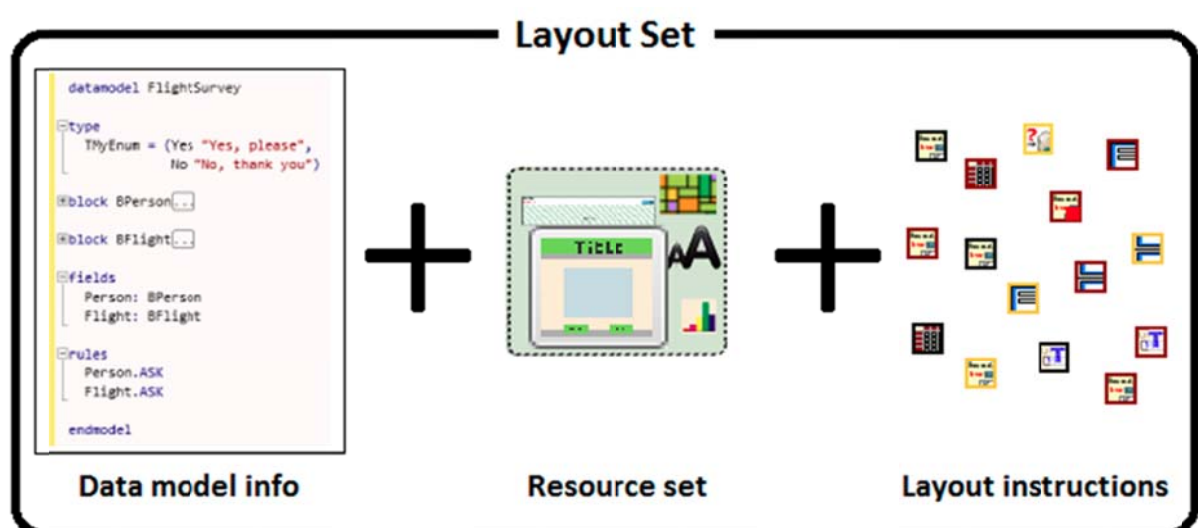
Will respondent candidness be affected by mobile devices? Although respondents are willing to reveal sensitive information about themselves in computerized self-administered surveys, it is unclear whether respondents will display the same level of candor when the survey is administered wirelessly on a tablet or a smartphone. Specifically is there potential for increased candidness given that with mobile devices respondents can more easily manipulate or shield them from view?

How much information can be displayed? Because of the small screens on mobile devices, it may be much harder to make all of the potentially useful information visible to respondents than it is with a laptop or desktop computer. Experiments on web surveys demonstrate that any information that respondents need to use should be immediately visible to them without their having to perform any action; the need for a mouse click or even a small eye movement may render information effectively invisible.

While the solutions to the challenges are ongoing the new Blaise 5 concept of Layout Sets structures the screen layout so that the optimum amount of information visible to respondents is presented in a visibly pleasing manner.

New Presentation Concepts

In a multimode survey, an institute might need a self-administered browser-based interface for web respondents, a much denser and more capable interface for its trained CAPI interviewers, and yet another interface altogether for mobile device users. Each user employs the same data model but has visual needs and possibilities related to their tasks. Each kind of user can be accommodated by using a different presentation known as a Layout Set.



A Layout Set consists of the blending of the datamodel source, a resource set, and layout instructions:

- **Resource Set(s)** – are a defined collection of font definitions, media, styles, texts and templates, that are standards intended to be used across projects
- **Templates** – define the layout of the content of a questionnaire: Master Pages, Field Panes, Grids, Input Controls, Navigation Controls, Panels, Property Parameter, etc.
- **Layout Instructions** – define page breaks, number of items per page, group instructions, critical items, etc.

Modifications

We were asked to replicate one of our corporate methodological surveys using Blaise 5. The survey was to determine the user's experience answering questions on different devices using different visual elements. We were tasked to utilize one set of code that could be run on specific devices (desktop, iPhone and iPad) in connected mode.

Most organizations will define the look and feel of a survey and the mode of collection. For each mode there will be a defined page layout based on organizational standards. These standards might establish the look of the company logo and name on a tablet vs. a smartphone. They might also include the number of questions on a page such as four on a desktop browser and one on a smartphone, as well as how one would navigate through the survey.

Field display modifications

Blaise 5 includes out of the box a number of field display layouts that can be used across devices. Then there are device specific layouts such as the navigation controls on a master page of a desktop browser versus a smartphone. These differences caused us to make basic modifications to the provided templates.

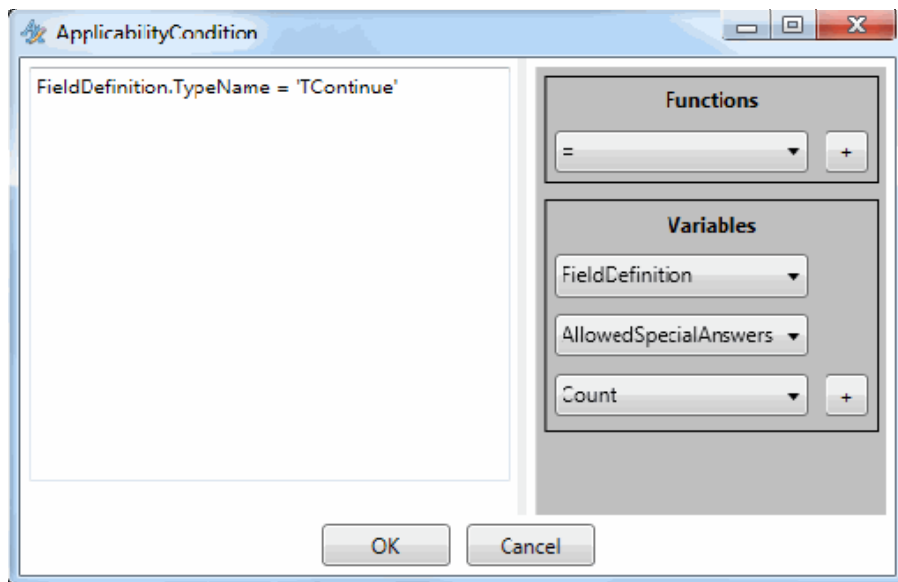
One of the modifications included the creation of user defined types that would determine the template to be used with fields of that type. Some of the user defined types we created were displays for currency, decimal currency, distance, and section introductions.

In addition to the basic modifications the specifications called for question displays to be changed between respondents. These included taking related questions and displaying them in a table format or one by one on a page, the display of questions with differing help aids, the number of questions on a page and response scales.

Some Basic Modifications

For Section introductions we created a User Type named TContinue to be used in the Applicability Condition of the QuestionText template. Next the Type Reference in the Resource

Database was updated with the type TContinue. The TContinue type reference then ties the datamodel fields to the Applicability property of a template where it is used in its Expression to automatically use the QuestionText template.



Another modification was to change the watermark to provide information as to what is to be entered and its format. We had to create a template, CurrencyReal, which includes a dollar sign and decimal. There were three ways to add a watermark *Dollars and Cents*:

- Using Text Roles
- Creating a user defined text in the resource database
- Defining a literal in the Watermark property of the template

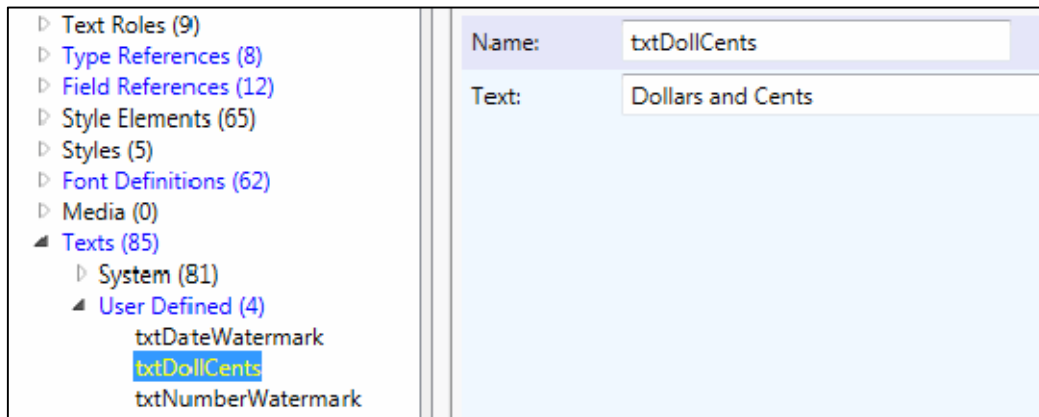
The Watermark text role is part of the initial Resource database and is linked to the datamodel through the ROLES section.

<code>roles = Help</code>	"The Help role provides a help button that the user can click to display question help text.",
<code>Watermark</code>	"The Watermark role provides a hint to the user as to what input elements are used for.",
<code>ToolTip</code>	"The ToolTip role provides a popup hint when the user hovers over a control."

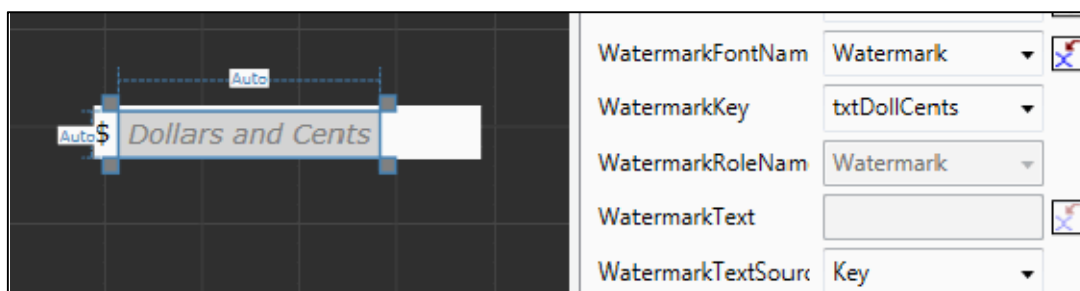
For the Gas question we added Watermark "Dollars and Cents".

```
Gas "How much do you pay per gallon of gas?"  
  WATERMARK "Dollars and Cents"  
  : TCurrencyReal
```

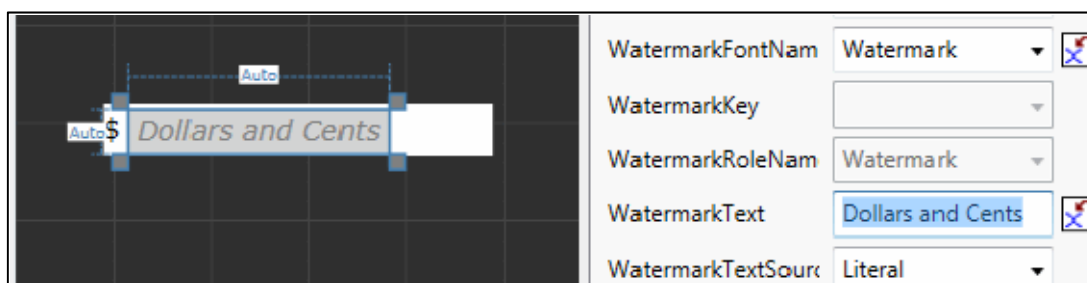
The second way is to add a User Defined text to the Texts in the Resource database. If there are multiple languages, this technique is the preferred way.



Each text has a name and is linked via the Watermark properties of the template.



The final way is to use a literal in the Watermark Text property of the template.



No matter which way you choose to implement the Watermark the result is the same.

Health Activities and Lifestyle Survey

Health Activities and Lifestyle Survey

How much do you pay per gallon of gas?

\$

[<Previous](#) [Next>](#)

[Save and Continue Later](#)

Technical Assistance: Blaise@Westat.com , x4172

Other Watermarks we used included the range for the year of birth and the number of characters for a string field.

Health Activities & Lifestyle Survey

What year were you born?

[<Previous](#) [Next>](#)

[Save and Continue Later](#)

Health Activities & Lifestyle Survey

Why do you say that?

Limit is 100 characters

<Previous Next>

Save and Continue Later

More Technical Modifications

When using groupings we found that the displays needed dramatic changes between devices. One change required the addition of a parameter to the Abreast template to allow for the vertical or horizontal display of a three part name depending on the device's layout set. The parameter was for the Arrangement property which determines whether items on a page are arranged in columns or rows.

Edit Parameters

Parameter Name	Type Name	Default Value	Control Name	Column/Row	Property Name
Width	Unit	Stretch	grid		Width
Arrangement	PlaceholderArrangement	1 Row	tableRows		Arrangement

Add... Edit... Delete

OK Cancel

The following two screen shots from the Layout Designer show the implementation of the Arrangement parameter.

What is your full name?

First

Middle

Last

Layout instructions

Table
Abreast

Parameters

Arrangement
1 Row

Width
Auto

What is your full name?

First

Middle

Last

Layout instructions

Table
Abreast

Parameters

Arrangement
3 Rows

Width
Stretch

In keeping with using one code set and different presentations we found the grouping of fields in a table while appropriate for running on a desktop or tablet would not be appropriate for an iPhone.

Health Activities & Lifestyle Survey

Health Activities & Lifestyle Survey

Please rate how serious a problem it is to be diagnosed with the following medical conditions.

	Not serious at all	A little serious	Somewhat serious	Serious	Extremely serious
The common cold	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Epilepsy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Emphysema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diabetes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Asthma	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hypertension	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High cholesterol	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lung Cancer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Influenza, or the flu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bronchitis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

<Previous Next>

Save and Continue Later

Technical Assistance: Blaise@Westat.com , x4172

Therefore, another technical modification we made was to create a new template to display Group fields one per page on an iPhone. Our objective was to add the Group text (Please rate how...) to each of the Group's question display (The common cold, Epilepsy, etc.). The first display of Group text would be in black and bold with the subsequent Group texts in gray per group question as a reminder of the general question to the respondent.

We created a new template from the Vertical field pane template. We modified the template by inserting a new row at top and adding a Label control. The Label control refers to a Field Reference that is mapped to a field in the datamodel containing the Group text. The resulting display follows.

Health Activities & Lifestyle Survey

Please rate how serious a problem it is to be diagnosed with the following medical conditions.

The common cold

☐ Not serious at all
☐ A little serious
☐ Somewhat serious
☐ Serious
☐ Extremely serious

Save and Continue Later

Page display modifications

One common sense change was to set one question per page for an iPhone layout set. Of course there are always exceptions such as the three part full name.

Navigation button display format can vary between iPad and iPhone. Buttons on an iPhone should be smaller due to the amount of display area. They should also be placed at the top the content area in order to be not covered by the popup keyboard. We created fingertip sized buttons for the Previous and Next pages.

Blaise App Things to Know

Blaise has been traditionally a one solution product running on the Windows platform. With the advancement and prevalence of mobile devices, surveys now need more than one solution thus making things more complicated.

Blaise 5 is entering into this complicated arena of more than one solution. Within this arena, Blaise has implemented native apps one for iOS and one for Android. With these implementations come integration issues that need to be worked through over the coming months. The following are an important subset of those issues:

- Device configuration, setup and security – protecting the device from unauthorized use
- Integration with a management system – for case deployment, tracking and auditing
- Data synchronization and encryption – transferring data from the device to the home office

Native Blaise Apps can run on tablets and smart phones either connected or disconnected mode. The following table provide by Statistics Netherlands explains these modes in more depth.

Run mode	Description
Thin client (Connected)	This is a client/server approach. Rules execution is done on the server. Data is also stored at the server. This is comparable to the internet version. Pages calculated on the server are sent to the app and displayed. In its turn the Blaise app sends a page response back to the server where a new page is calculated based on the response. The device must be online in order to run in this mode.
Thick client (Connected)	This is a hybrid mode. The survey definition must be downloaded to the device, because the rules execution is done on the device. Data retrieval and storage is done at the server. The device must be online during retrieval and saving of the data. This mode is faster than thin/client, because there is no need to visit the server between pages.
Thick client with fallback (Connected)	Same as thick client, except for the situation in case there is no connection during the save of interview data. If this is the case then the data will be stored on the device and needs to manually be uploaded once the device is online again. The app has functionality to upload the data manually.
Disconnected	Rules execution and data storage is done on the device. The survey definitions must be downloaded first before the survey can be conducted. Collected data will be stored on the device and must be manually uploaded to the server. The device does not need to be online when running this mode is being used.

In addition to the information provided in the table there are a few more things to know.

In disconnected mode:

- The rules engine that runs on the device has the same functionality as the one that is used for Windows DEP and Internet.
- You will need an iOS device that is running iOS version 5.1.1 or higher. Android needs to be 2.1 or higher. So, you can use relatively old devices with the apps, although newer is better of course.
- In Server Manager you have to set Run mode to the disconnected or thick client. You find this setting at the Server Park level. The apps will then make available a download button that lets you download and install the survey files on your device.

In connected and disconnected mode:

- Both the iOS and Android apps don't use a browser at all, they render the layout themselves directly on the device's screen. There is no difference in rendering the layout for the disconnected and connected mode.
- Client-side rules are implemented in the Blaise apps, to avoid server roundtrips.
- The default presentation on a mobile device for enumerations with a dropdown layout is a spinner.

The 2007 Blaise NG White Paper and the Blaise 5 Result

Mark M Pierzchala / Mark M Pierzchala, MMP Survey Services, LLC (MMPSS)

1. The White Paper and the Working Group

The Blaise Corporate License Users Group (BCLUB) gave guidance for Blaise Next Generation (Blaise NG) through a document called the White Paper. The BCLUB understood that Blaise Next Generation would include many major changes. While Blaise 4 was (and still is) a very successful system, Blaise NG would have to adapt to a demanding and rapidly changing environment. The BCLUB wanted to define a system that would build on the strengths of Blaise 4, add features, and anticipate future needs.

The BCLUB formed the White Paper Working Group in early 2007. It included 13 people from 8 countries and 10 BCLUB members. These individuals had numerous years of Blaise and survey experience. They came from 9 National Statistics Offices and 3 private firms. Additionally, three other BCLUB institutes contributed strategic requirements. Mark Pierzchala from Mathematica Policy Research, Inc. was named chair.

The Working Group started work in May 2007 and completed in September 2007. The White Paper was discussed at the BCLUB meeting on September 25, 2007 during the 11th International Blaise Users Conference. The White Paper is 45 pages long. The group also produced a 65-page supplement that gives a record of technical discussions. A few years after the White Paper was published, The Blaise Team changed the name of the system from Blaise NG to Blaise 5.

The BCLUB formed a follow-up group in 2012 to revisit a few topics due to changing technological developments. The BCLUB continues to meet and discuss Blaise 5 developments at least once a year. The White Paper remains confidential to the BCLUB and Statistics Netherlands.

All opinions in this paper are of Mark M Pierzchala. He visited Statistics Netherlands in February 2015 and talked with the Blaise Team about this paper's main points. Blaise Team members also reviewed a draft of this paper for technical accuracy and to make sure the plans of the team were correctly stated.

2. The White Paper Contents and Guidance

After an Executive Summary and an Introduction, the White Paper continued with three interesting sections: (1) The Use of Blaise Today, (2) The Strengths of Blaise, and the (3) Scope of the Blaise System. These sections demonstrated the breadth of Blaise 4 and the magnitude of the conversion task. It showed that other systems would have a difficult time duplicating its capability.

2.1 The Use of Blaise Today

This section told of the many uses of Blaise. It listed survey areas such as social, labor, economic, agricultural, health, expenditure, and other topics. It listed kinds of surveys including household surveys with multi-level rostering of people, person surveys, institutional surveys, repeated surveys, longitudinal surveys, programs of surveys, studies that combined several surveys, and mixed mode surveys.

2.2 The Strengths of Blaise

This section described what fundamentally separates Blaise 4 from other systems. It mentioned a very strong programming language with many metadata elements. It explained how the checking mechanism (at once global and selective) enables great performance and flawless navigation. It demonstrated how the split-screen interviewer interface was innovative and effective. It showed how the system architecture enables huge questionnaires, and how its structure allows for clever solutions for difficult problems.

2.3 Scope of the Blaise System: Existing 4.x Tools

This was an elaborated listing of the 26 modules of Blaise 4. It showed that the system possessed modules far beyond an interviewing program. It gave a sense of the size of the development task. Some additional modules have been added since 2007 such as Computer Audio Recorded Interviewing (CARI).

2.4 Technical Direction

This section forms the heart of the paper. It covered 13 technical areas including:

- Platform
- Metadata
- Data
- Layout
- Case Management System
- Mixed Mode
- Development Environment
- Language Enhancements
- Integration and Interoperability
- Documentation
- Migration
- Performance
- Security

Each area gave a summary of guidance points and background explanation. The companion document provided extra details on most points.

2.5 Appendices

Appendix A of the White Paper listed Working Group members and told of their varied backgrounds and areas of technical expertise. Appendix B gave a high-level overview of the .NET environment.

2.6 Guidance, not Instructions

The White Paper gave guidance to Blaise 5 development, not strict instructions. It acknowledged that the Blaise Team would have wide discretion in Blaise 5 development and that for some items, development decisions would differ from the White Paper guidance.

3. Evaluating the White Paper almost 8 Years Later

The White Paper Working Group knew that predicting the future would be difficult. It knew that technical innovation was proceeding very quickly and that platforms would spring up overnight. Nevertheless, it

did its best to anticipate technological change, and overall did a decent job. There are however, a few things that the White Paper missed or underestimated.

3.1 Accessibility

The White Paper said nothing about accessibility and usability for disabled people. This was a major omission. Even in 2007, there was much discussion for the need to make self-administered surveys accessible for the blind, the visually impaired, and those with motor impairments. This omission was noted in 2010 by The Blaise Team and Blaise 5 is being developed with accessibility requirements in mind. The Blaise Team also back-fitted accessibility features into Blaise 4 in 2013.

3.2 The Proliferation of Mobile Devices

Smart phones and modern-day tablets were just emerging in 2007 so the Working Group knew about them. However, it did not estimate (who did?) how quickly these devices would become mainstream. The idea that a respondent would answer a web survey on a cell phone was not considered, or that an interviewer might use his or her own device to conduct an interview. The paper suggested that The Blaise Team investigate how Blaise NG might run natively on a mobile platform; now this capability has become an unavoidable requirement for iOS and Android devices running Blaise 5. Additionally, the White Paper said nothing about *time to release* of new versions.

4. How Blaise 5 Development compares to White Paper Guidance

It is possible to compare the still-developing Blaise 5 against the White Paper guidance. The purpose of this section is to state where those differences are, and the reasons for these differences. Sometimes, the emerging Blaise 5 system has not yet implemented an item but plans to do so in the future. Note that the Blaise Team continues to consult with the BCLUB on priorities.

4.1 Platform

Blaise 5 tracks this section closely. This new Blaise 5 version is the first to operate its runtime modules (e.g., interviewing) not only in Windows, but in non-Microsoft environments such as iOS and Android mobile devices. It can use Linux as a web-survey host for the first time. These multiple runtime environments have proved a challenge, not only because of different operating systems, but because of screen size, pixel density, and different keyboard layouts and available keys.

4.2 Metadata

In Blaise 4, a LAYOUT section in the source code language gave layout instructions that governed page breaks and the use of templates from the Mode Library. The Working Group explicitly stated that the LAYOUT section should stay in the source code. However, the LAYOUT section is no longer valid for Blaise 5. In its place is a Layout Designer in the Control Centre where the developer can manually insert references to templates and page breaks. This is a major departure from the Working Group direction. The Layout Designer is difficult to use for large surveys due to performance issues (which will be fixed) and the number of manual actions that have to be performed. To address this issue, the Blaise Team will design a Layout Language that will replace the LAYOUT section. (See 4.4 on Layout.)

The White Paper suggested that the Blaise Team investigate the possible creation of a Graphical User Interface (GUI) for interactive statement of metadata. The Blaise Team made some investigations but discontinued work for now. Since it was mentioned as a secondary priority by the White Paper, the Blaise Team should not spend any time on this at the moment.

In Blaise 5, it should be possible to associate descriptive text with an assignment statement. This is not yet possible.

4.3 Data

The Working Group wanted to get away from the proprietary Blaise 4 database in favor of an open source solution such as MySQL. The Blaise Team chose SQLite which satisfies the suggestion. However, the Blaise 5 storage within the SQLite database is still proprietary and not of a RDBMS format. A mapping to such a RDBMS format can be achieved in Blaise 5 with the .bdix file. This is much like the .boi file solution in Blaise 4. This choice was made by the Blaise Team in order to minimize data storage size (the typical RDBMS being a model of bloat) and to achieve performance goals (especially field-to-field interviewer performance).

An extended range of field statuses in addition to EMPTY, DK, and RF was directed by the Working Group. This is not yet done but the Blaise Team will implement it soon.

BCLUB members wanted more flexibility with a more robust (or forgiving) database definition. For example, if a field range changes from 0..1 to 0..7 the database checksum should not change. That is, minor changes in field definition should not result in incompatible data files that require a data update procedure. The Blaise Team notes that there are unanticipated problems with this White Paper suggestion. These include the situation where you mix databases with different field definitions and it is not clear what should happen in these cases. This can still be a BCLUB discussion item.

Neither the White Paper nor the Companion Document explicitly mentions the phrase client-server. There are several mentions of 'server' in the context of web surveys, and several mentions of 'client'. But you never see client-server as a requirement. (Some BCLUB meetings may have talked about the client-server paradigm.) In Blaise 5, client-server is a required paradigm. Practically speaking, this means that the database must be installed (like Blaise IS in Blaise 4), not just copied into a folder. This also means that the Blaise 5 database is stored in a particular folder location. There are a few awkward effects of this choice. For example, institutes often have external files stored in a sub-folder of the main folder, and this is not yet possible. Some more work needs to be done to give needed folder flexibility.

4.4 Layout

The White Paper section on layout says all the right things. New capabilities are needed above those provided by Blaise 4 because of new devices, screen sizes, screen orientations, and screen resolutions. It also mandated that the Blaise Team find a unifying layout concept that would allow the efficient and flexible generation of screens. In hindsight, this section vastly understated the complexity of achieving the needed layout flexibility. The Blaise Team has spent enormous amount of time achieving the goals of this section. It came up with the Resource Database to replace the Mode Library. It came up with the Layout Designer to (manually) replace the LAYOUT section.

The Working Group specifically requested in this section that the LAYOUT section in the source code remain, and that the generating of screens based on settings and key words should be maintained. The reason the Working Group wanted to maintain the LAYOUT section is because the generation of pages in Blaise 4 is extremely efficient. Some Blaise instruments have tens of thousands of pages. The Working Group did not want to have to manually adjust layout. This suggestion was not taken for Blaise 5. Furthermore, while the LAYOUT section was removed from the source code, the GROUP concept was introduced into the source code (a Working Group suggestion). Maintaining the LAYOUT section in the source code seems entirely consistent with placing GROUP in the source code.

The Blaise Team deems the LAYOUT Section too limited for modern-day challenges, and that a more powerful layout language is needed. Additionally, there can be more efficient use of the Layout Designer such as generating and displaying only one model page instead of many identical pages. At the time of this writing, layout possibilities in Blaise 5 are far more flexible than in Blaise 4. It is also very time consuming and difficult to deal with the Resource Database and the Layout Designer. The Blaise Team has committed to make layout much easier to achieve in Blaise 5. In the end it doesn't matter how this is accomplished as long as developers can efficiently create layouts across modes and devices.

The information-rich split-screen interface, and all of its ease-of-use operability, was also required by the White Paper. This is in progress.

4.5 Case Management System

At the time of this writing, development of case management in Blaise 5 has not yet started. The Blaise Team has been gathering requirements from BCLUB members for this section. The CATI call scheduler in Blaise 5 will be like the one in Blaise 4. Blaise 5 will include a multimode survey management scheme.

4.6 Mixed Mode

Three aspects of mixed mode surveys are mentioned in this section. First is the ability to generate different kinds of screens for different modes. This has been achieved. Second is the use of settings to achieve some multimode capability. This has been achieved.

Third, however, is the ability to handle field attributes and some rules differently between modes, in a more elegant way than in Blaise 4. For example, in an interview, if a respondent does not know the answer to a question, the interviewer should be required to enter a DK through a function key. On the other hand, in a self-interviewing situation such as a web survey, a more common practice is to allow the respondent to pass over the question without answering. The non-answer then is interpreted as a Don't Know or a Refusal answer in the downstream analysis. These differences in mode conventions can be done in Blaise 4 through a lot of programming. A much more elegant solution would be to allow field-level differences between modes. Blaise 5 should be told how to implement DK and RF differently between modes at the instrument level. The Blaise Team is currently implementing this capability.

4.7 Development Environment

Blaise 5 has a fully capable development environment now. It has a few large features that Blaise 4 does not have which are the Layout Designer and real-time parsing. The latter can be turned off if working on an extremely large instrument.

The White Paper talked about a GUI interface to enter metadata, but this was not a priority. This GUI has not been realized. At this point, it does not seem like a GUI module for core metadata would be worth the effort, and any such an effort would distract from other higher priority items.

4.8 Language Enhancements

More than twenty language enhancements were listed in the White Paper. One of these, the request for indefinite arrays, has proved difficult due to page layout issues, but it is still on the to-do list. A few others, having to do with table display and operability (e.g., a vertical table), are really layout features. In fact, it is now possible to do a vertical table in Blaise 5 by defining a template in the Resource Database.

The White Paper mentioned the introduction of GROUP. This would allow an easier way to combine 2 or more fields that visually or otherwise should be combined. A classic example is the idea of a quantity / unit construct, for examples distance travelled (number followed by either miles or kilometers). Groups have been added to Blaise 5 and form the basis for all multiple-item displays including tables.

A few of the proposed language enhancements affect the core of the system and thus would have to be done within the Blaise 5 language. These include extended field attributes for multimode surveys and user-defined field attributes. The Blaise Team is currently working on these core improvements. The implementation of these features will considerably ease the development of multimode instruments.

Blaise 5 has a new construct called ROLES. This was explicitly mentioned in the White Paper and is a promising new language feature that helps satisfy some auxiliary requirements. For example, in Blaise 4 it is common to use LANGUAGES to state metadata items such as SAS variables, or to construct item-level help. ROLES now fulfill this need and LANGUAGES are reserved for spoken languages.

The LANGUAGES section in Blaise 5 is much more powerful than in Blaise 4 with the explicit inclusion of all known written languages. Blaise 4 did not handle Asian languages well and struggled with Right-to-Left languages. A demonstration instrument about commuting habits is on the Blaise website. It includes 10 languages including Arabic, Chinese, Dutch, English, French, Greek, Hebrew, Hindi, Japanese, and Spanish. A language control allows you to switch between the languages on the fly.

Some of the White Paper language suggestions are really best implemented as model blocks of code. These may be provided through the use of the SNIPPET capability. This is a new Blaise 5 feature that allows specification of some constructs. You could imagine a SET-most important-second most-important-construct to be modeled as a snippet.

Many of the suggested language improvements would reduce the amount of programming we now have in Blaise 4 to implement some features. These include randomness and rotation, SET-most important-second most important constructs, and others. The desire by the Working Group was to have explicit features in the system to simplify these more demanding requirements. Some of these are still under consideration by the Blaise Team. This especially includes randomization; others are not a high priority at the moment.

A few of the language suggestions such as some string manipulation functions, should be relatively easy for the Blaise Team to implement, but it is a matter of how high a priority this would be.

4.9 Integration and Interoperability

The White Paper mentioned three items: (1) an Application's Programmers Interface (API), (2) a services approach to components, and (3) alien routers. All three of these suggestions have been implemented, including alien routers in early 2015.

4.10 Documentation

This is about an instrument self-documenting feature. In Blaise 4 there is a system called Delta that can generate views on the metadata including a flow chart. A parallel system created by the University of Michigan Survey Research Center called Michigan Questionnaire Documentation System (MQDS) also provides various views on the instrument. Both Delta and MQDS generate their electronic views from the compiled metadata file of the Blaise 4 instrument. Blaise 5 will not have a Delta capability. It does have a metadata API that would allow a third party such as the University of Michigan to recreate their MQDS system for Blaise 5.

Blaise 4 also has the Cameleon metadata language that allows various descriptions of FIELDS, such as a data dictionary or SAS description. Blaise 4 also allows metadata access by Manipula. SAS and Stata scripts are programmed in Blaise 4 Manipula for example. There are no plans to duplicate Cameleon in Blaise 5. Manipula in Blaise 5 will continue the metadata access. This more than replaces Cameleon. The Blaise 5 metadata API will also be available.

4.11 Migration

The White Paper spends quite a bit of space discussing the migration of Blaise 4 instruments and surveys to the Blaise 5 environment. Given the need for a complete restructuring of the Blaise system, no one believed that you would be able to press a button and generate all Blaise 5 changes.

Blaise 5 has a utility that can convert most Blaise 4 source code to Blaise 5 source code. While this capability will greatly ease the conversion of Blaise 4 to Blaise 5, it does not complete the task. There are too many differences between Blaise 4 and Blaise 5 to completely escape manual modifications to the source code and configuration files. This is mainly due to the layout and platform demands of the modern-day computing environment.

Among the Blaise 4 features that will disappear or change in Blaise 5 are the use of environment variables, different kinds of alien routers, tables (become groups) and how layout is accomplished. An upcoming Institutes Guide to Blaise 5 will thoroughly discuss these topics and explain how to efficiently move from Blaise 4 to Blaise 5.

4.12 Performance

The Working Group specifically mentioned performance as a reminder that one of the great attractions of Blaise 4 was its item-to-item fast speed, especially with large interviewing instruments. Performance for Manipula on data files, large numbers of users accessing a web site, and so on, was also mentioned. As Blaise 5 emerges into production, performance will be easier to measure. The Blaise Team is aware of some performance issues and will solve them through optimization analyses and procedures. BCLUB members can assist with their own performance testing.

4.13 Security

The Working Paper had only two requests for this topic. One was the ability to encrypt data on a field-level basis (e.g., for social security numbers) and to be able to work with third-party systems to be able to encrypt data on a database level.

The native Blaise 5 database storage encrypts every field automatically, except that the key fields are exposed. Also, by using the SQLite database as a default, third party solutions should also work.

5. Summary

The development of Blaise 5 has turned out to be very challenging. The Blaise Team has spent a large amount of time on layout concepts and platform in order to meet the challenges of representing a Blaise questionnaire on any device. This has caused them to deviate at times from The White Paper suggestions or to implement the suggestions in a different way.

Blaise 5 development is still underway. Some White Paper suggestions that are not yet implemented will come soon. Following is a discussion of where Blaise 5 should more closely meet the White Paper.

5.1 Layout

The Blaise Team has done a very good job of defining layout concepts that work across devices and modes. They allow the institute to define their own templates in the Resource Database and to implement them in an instrument through the Layout Designer. They have built a solid foundation.

At the current time however, layout is hard to achieve in Blaise 5. This has been noted by several early adopters. The use of the Resource Database requires a large learning curve, as does the Layout Designer. Even for an accomplished user, the manual insertion of layout elements requires too much time.

The addition of a Blaise 5 Layout Language to replace the Blaise 4 LAYOUT section will be a high priority. The Blaise Team should improve the performance of the Layout Designer and a better default Resource Database model should be developed. This will allow institutes to modify an existing solution rather than define one from scratch. Additionally, some advanced techniques to generate layout should be documented (these will be published by the end of 2015 with the Institutes' Guide).

Related to layout are operability aspects of the Blaise 5 runtime. Blaise 5 should be at least as easy to use as Blaise 4 and the item-to-item performance should be the same for interviewers.

5.2 Language Enhancements

Blaise 5 retains the powerful Blaise 4 language with key improvements. Even with some changes (e.g., with GROUPS) the Blaise 4 programmer will have little problem with the Blaise 5 language.

The most important language enhancements yet to be implemented relate to multimode features. This is especially true of mode-specific attributes that would allow instruments by default to more easily handle the different practices between self- and interviewer-administered questionnaires (see 4.6 above).

5.3 Migration

Blaise 5 provides some migration tools, but the differences between Blaise 4 and Blaise 5 require substantial manual adjustments. These fall into the categories of (1) instrument migration, (2) deployment, (3) integration, and (4) performance. These are very practical concerns and impact the daily fielding and operations of an ongoing survey.

Blaise 5 has a very good on-line help. There are 6 tutorials that help with layout and deployment. More documentation is planned and underway. There are now a few good instrument examples, and many smaller datamodels that demonstrate a feature. An Institutes' Guide is planned for end of 2015 that will explain how to address many of the day-to-day procedures that an institute must execute.

6. References

Pierzchala, Mark M., et al. "Blaise Next Generation; A White Paper by the BCLUB Technical Working Group on Blaise NG." Presented to the BCLUB at the 11th International Blaise Users Conference (Survey Research Center, University of Michigan), Annapolis, MD, September 2007. (Available to BCLUB)

Pierzchala, Mark M., et al. "Blaise Next Generation; Companion Document, A Record of Discussion of the BCLUB Technical Working Group." Presented to the BCLUB at the 11th International Blaise Users Conference (Survey Research Center, University of Michigan), Annapolis, MD, September 25, 2007. (Available to BCLUB)

7. Appendix – List of the Members of the Working Group

Mark Pierzchala, US MMP Survey Services, LLC, in 2007 with US Mathematica Policy Research, Inc.

Karen Bagwell, US Census Bureau

Gina-Qian Cheung, US University of Michigan

John Kinder, UK Office for National Statistics

Roger Linssen, the Blaise Team, Statistics Netherlands

Rebecca Lui, Statistics Canada / Statistique Canada

Grayson Mitchell, Statistics New Zealand

Sylvain Moreau, Institute National des Statistiques et des Etudes Economiques (France)

Zsolt Papp, Statistics Hungary

Juanita Pettit, Australian Bureau of Statistics

Mike Rhoads, US Westat

Roger Schou, US National Agricultural Statistics Service

Jeroen van Velzen, User, Statistics Netherlands

Back to the Future: Converting NCVS to Blaise5

Daniel Moshinsky, US Census Bureau

Purpose and scope of this paper

The primary objective of this test is to examine the functionality of the instrument under Blaise 5. Can we collect all the same data we collected in Blaise 4? What modifications, if any, would we need to make to our Blaise syntax, our instrument design, or our data collection procedures?

Questions about survey management, deployment capabilities, and data storage were outside of the scope of this test. The paper will touch on issues related to the layout and data storage, but it is not our central focus presently.

This test was intended solely to test a CAPI survey running on a Windows laptop. Thus we made no attempts to run the converted survey on other mobile devices.

It is understood that there are many features of Blaise 5 that have not been implemented at all, or that have not been implemented fully. Nevertheless, it is worthwhile to explore the capabilities of the Blaise 5 system both in order to provide developers with valuable feedback, and most importantly in order to better get to know the system.

What is NCVS and how does it work in Blaise 4

The National Crime Victimization Survey (NCVS) is a monthly household crime survey. It was first fielded as a Blaise 4.6 CAPI survey in 2006, and has since been converted to Blaise 4.8.4. Every household member over the age of 12 is asked about crimes that happened to them in the last 6 months.

The NCVS instrument initially conducts a “Screener” where the respondent is asked about types of crimes that happened (such as theft, assault, car theft, etc.), and about the frequency of each type of crime. Once the screener is complete, the “Incident Report” section is asked. Here the respondent is asked a long series of questions about details of each reported crime. For example, if R1 reports that his car was broken into 2 times, and he was assaulted 3 times, R1 will be asked to provide 5 Incident Reports.

After the Household Respondent’s interview is complete, the rest of the household is interviewed in a similar fashion.

Due to the size of the instrument and the nested looping involved, it was not possible to program the instrument, in Blaise 4, in a straightforward fashion of looping through each person’s incidents. We ran into page limits, block size limits, and performance lags. As a result, we implemented an “ask, store, empty, and re-ask” approach presented at the IBUC conference in Annapolis, 2007 (“*Challenges in Converting the National Crime Victimization Survey to Blaise*” Charlie Carter, Roberto Picha, Michael Mangiapane, Alexis Arrington and Daniel Moshinsky; US Census Bureau, USA).

In brief, this approach asked the screener questions in a non-arrayed block. It then created a rostered “storage” block of the same type. When the screener questions were completed, the values were copied into the “storage” block. Similarly, there was also one non-arrayed Incident Report block. After each incident report, the block was copied into the arrayed “storage” block along with linking keys. Then, the incident report block would be emptied and come back on route, and the process would repeat until there were no more incidents. When the interview with the respondent was complete, both the screener and incident report blocks would be emptied to prepare them for the next person.

This approach resulted in substantial improvement of run-time performance.

Why choose NCVS for the conversion test

The Census Bureau has been investigating Blaise 5 since its earliest test releases. Now that Blaise 5 is approaching maturity we wanted to implement a full conversion of one of our established Blaise 4 CAPI surveys into Blaise 5.

NCVS was chosen because of its complex Blaise language functionality and large size. We have previously successfully converted small surveys in our test environment, but this time we wanted to test the limits of Blaise 5 to see if we would uncover issues for Statistics Netherlands to address as it continues to develop the software.

In addition, NCVS does not heavily rely on the use of Manipula. Since Manipula is not yet available under Blaise 5, it was appropriate to pick a survey that is not reliant on it.

Conversion process

The specific steps in converting a Blaise 4 instrument are described in the Help documentation. For this paper, just the general steps are outlined, along with a few suggestions.

First, all components of the Blaise 4 instrument were converted to the latest available build of Blaise 4.8.4. In our case, it was 4.8.4. Build 1888. Next, the source code must be converted and prepared in Blaise 5 in order to generate a compiled datamodel. It was important to first make sure that the Project Manager in the Blaise Control Center included all the files associated with the project, including the modelib, and all the source search path locations were listed (without spaces) in the Blaise 4 ->Blaise 5 source converter utility.

The .bpf project file is then converted into a Blaise 5 solution (.bsol). Several issues had to be corrected in order to successfully compile the instrument under Blaise 5.

After a .bmix file is successfully generated, the test input data contained in our Blaise 4 database had to be converted to a Blaise 5 database. This Blaise 5 database will provide the input data needed for testing the NCVS instrument. Manipula was used to load test data (sample control values, address information, and existing information about the household) into the Blaise 4 case

database. We used the database conversion process to load the test data from the Blaise 4 database into our Blaise 5 database.

The converted .bdix and .bdbx files were placed in the location specified in the “Deploy Path” under File -> Options. Overall, the process of converting the source code and the main and external databases for this survey was relatively painless. Only minor modifications, that were self-explanatory, needed to be made.

Sets and Enumerated Lists

When answer categories are converted, answer category numbers do not display by default. The easiest way to display answer category numbers is to “Move Up” SetCodeText and EnumeratedCodeText in the list of Category Templates, such that SetCodeText is above Set, and EnumeratedCodeText is above Enumeration. Templates are applied in the order they appear on the list, if they meet the Applicability conditions.

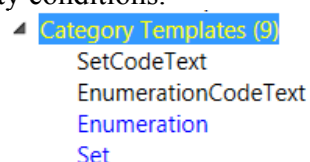


Figure 1. Re-ordered Category Templates

In addition, it may be necessary to change the default templates for answer lists. To do that, go to Input Controls > Data Value Templates > Answer List, and, in Properties, click on “DefaultTemplates” and add Category (EnumerationCodeText) and Category (SetCodeText).

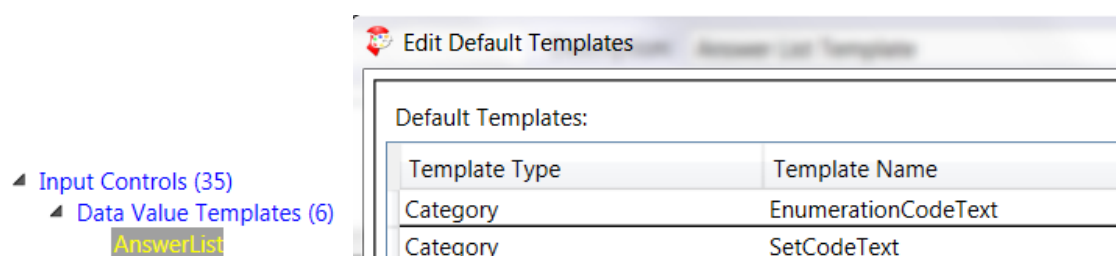


Figure 2. Setting the default template to use CodeText

After these changes, the answer category numbers will appear:

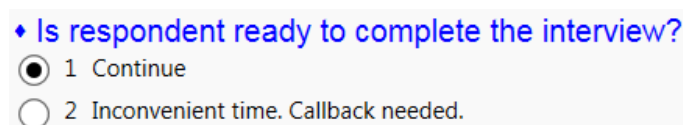


Figure 3. Numbers now appear next to answer categories.

Testing approach

Several testing scenarios were keyed with Blaise 4 and Blaise 5 instruments running side-by-side in order to test routing, layout, and performance issues. After each data entry session, the Blaise database was checked for the expected values of various significant fields. Specific issues uncovered during the testing are discussed in the sections that follow.

Each time a new release of Blaise 5 was made available, we attempted to re-test, as much as possible, the issues discussed in this paper. The latest available version at the time of this writing is 5.0.2.721.

Blaise 5 Instrument Testing Results

Attribute EMPTY, DONTKNOW not allowed for key field

Several of our external lookup files allowed for EMPTY in key fields (for example, CountryCode). Also the datamodel-wide setting was to permit DONTKNOW. This rule has been tightened in Blaise 5, and a compile-time error will result. So, if CASEID is your primary key, you will need to give it an explicit NODK attribute.

Compare functions

One of the biggest sources of generated warnings – as well as incorrect run-time behaviors have to do with a change in how compare functions, such as ‘=’ and ‘<>’ operate in Blaise 5. In fact, over 100 warnings were generated:

▼	176	16	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	145	16	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	293	63	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	340	3	Layout section is ignored.
▼	125	19	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	162	57	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	162	55	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	164	33	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	165	37	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	189	48	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	205	29	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	321	22	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	331	41	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	337	83	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	144	50	The meaning of this expression differs from Blaise 4: It is only 'false' when the left hand side has been assigned the value '0'.
▼	250	21	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	284	36	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	580	17	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	280	49	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	303	19	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	316	36	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	393	19	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	504	76	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	598	25	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	613	71	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	113	14	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	117	16	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	145	38	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	145	79	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.
▼	171	37	The meaning of this expression differs from Blaise 4: it is only 'true' when the left hand side has been assigned the value '0'.

Figure 4. Expression warnings.

According to the Help documentation:

The compare functions '=' and '<>' consider values to be different from empty and DK and RF. (In Blaise 4.8 Empty, DK and RF are equal to the default value of the type of the field. Exception: an enumeration category with code 0 is not equal to empty). An empty string '' is unequal to the special value EMPTY.

When inspecting these warnings, it was difficult to tell which of these created a potential instrument issue, and which could be ignored. For example, many of the warnings were caused by the line:

```
IF mode = '0' then...
```

The value of “mode” is filled from the input file. It is a field of type STRING[1], and it can be set to ‘0’. Thus comparing the value to ‘0’ is perfectly appropriate.

Other instances of this warning message, however, did point to a real problem. For example:

```
if Roster[I] = '' then ...
```

Warning - The result of the expression is always 'false'. Use = EMPTY instead.

In this case, we are looking for a member of the Roster string array that is empty. Previously the blank quotation mark notation worked, but in Blaise5 you must use `Roster[i] = EMPTY`.

This type of notation (and its opposite, `Roster[i] <> ''`) was present throughout our instrument and was the cause of many instances of incorrect instrument routing in Blaise 5. Similarly,

```
...AND harmedFlag = 0 THEN...
```

relied on an implicit value of 0. In many instances this had to be modified to `harmedFlag <> 1`. In the following example, field `LOSTPAY` was coming on route when not expected:

```
IF (DAYSLOSTWORK <> 0) THEN  
    LOSTPAY ...
```

`DAYSLOSTWORK` is off-route with an empty value. In Blaise 4, this resulted in the comparison `DAYSLOSTWORK <> 0` evaluated to “false” and the `LOSTPAY` field was skipped. But in Blaise 5, `LOSTPAY` was coming on route.

Modifying the syntax to `IF (DAYSLOSTWORK > 0){changed to greater than}` corrects the issue.

“Self-Reference”

Reusing field names

In Blaise4 we had an array called “count_down[1..30]”. An element of this array was passed to a sub-block. The received parameter (ungracefully) used the same name:

```
import count_down : integer
```

This worked in Blaise 4, because the rules engine referenced the local name first, and so any reference to count_down in the sub-block was understood to be a reference to an integer, and not to the array declared at the higher level.

In Blaise 5, however, this appears to be reversed, resulting in an error message when trying to pass this integer as a parameter:

```
Sect09 ( {Import} count_down)  
Error - Cannot (implicitly) convert 1st parameter of type ARRAY[1..30] OF 0..30 to  
type Number
```

The solution is to rename the local variable.

Emptying a block

As previously described, emptying blocks in order to bring them back on route is a critical feature of the NCVS instrument design under Blaise 4. It is the mechanism with which the instrument accomplishes multi-layered looping.

However, we discovered that assigning a block to EMPTY is not currently possible in Blaise 5. It has been promised that the functionality will be added in future releases.

This limitation also affected the ability to create a Replacement Household – to empty the household roster when all the people who lived there during the last interview wave have moved out.

To work around the lack of the block emptying functionality, we emptied just certain key fields in the beginning and end of the blocks in order to enable us to cycle through the entire household.

Layout Issues

We are still in the process of experimenting with various screen layout options under Blaise 5. We would like to continue to run our surveys in keyboard-driven environments that our interviewers are used to, while also supporting touch screen interactions.

Fonts

Blaise 4 font tags (e.g. @F, @\, @\)) are automatically converted into Blaise 5 tags. For font tags, this happens if corresponding user-defined fonts are defined in the Resource Database:

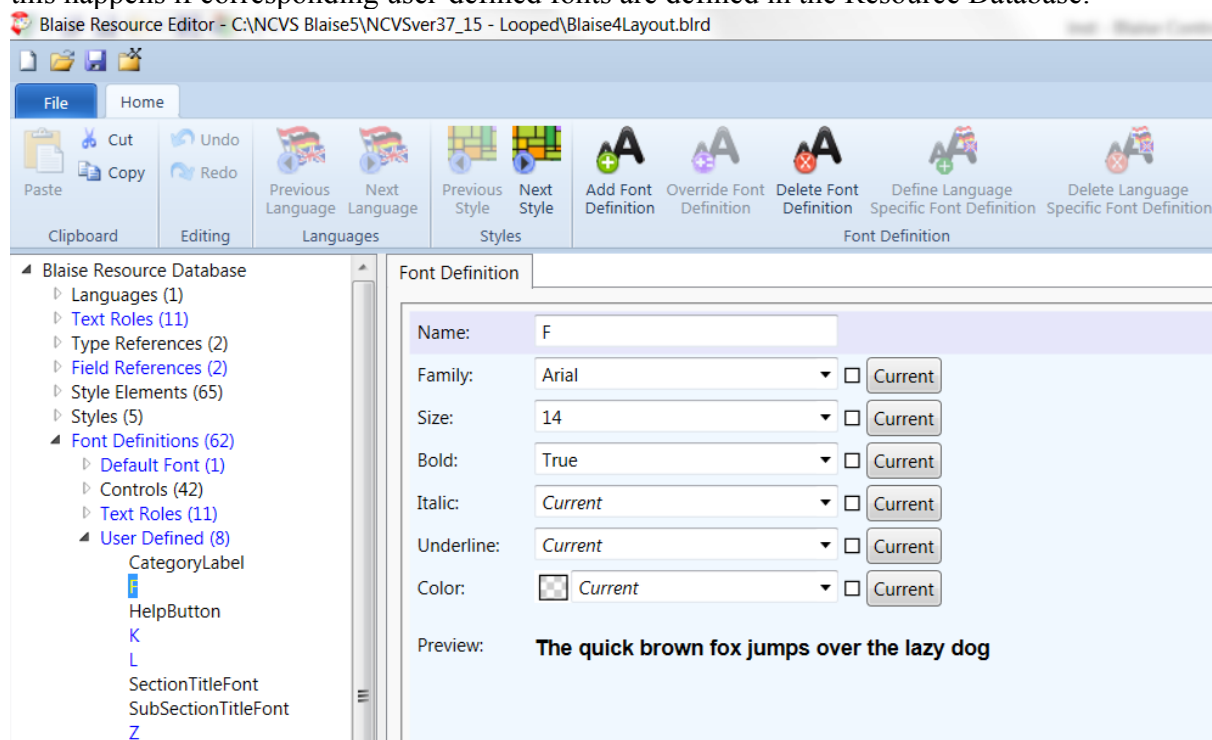


Figure 6. User Defined font definitions

In the screenshot above, you can see the font definitions that will result in tags <F>, <K>, <L>, and <Z> in our question text.

Thus, in Blaise 4, text for the field LOCATION is defined as follows:

```
LOCATION ("LOCATION")
  "@/@|@Zs@Z@L Ask or verify:@L
  @/@/@/@|@FWhere in an open area, on the street, or on public transportation
  did this incident happen? ^displayRoster@F"
```

In Blaise 5, the text definition looks like this, with the @ tags converted to <> brackets:

```
LOCATION ("LOCATION")
  "<newline><tab><newline><z>s</Z><L> Ask or verify:</L>
  <newline><newline><newline><space count=4><F>Where in an open area, on the
  street, or on public transportation did this incident happen?
  ^displayRoster</F>"
```

Fill fonts

Now, what happens if the field contains a fill and there are font tags inside of that fill? As it turns out, Blaise 5 will not interpret those font tags, and will instead display them on the screen, as seen in this example:

The string you entered is aa. Formatted it looks like <L>aa</L>

Figure 7. Font tags inside this fill are not interpreted.

To prevent text enhancement tags from appearing literally in the text, the variable text should be placed between curly brackets (e.g. `^{MyFill}`). Text enhancements will then result in enhanced rich text. It is not clear what the purpose of this syntax change is because it seems to me that you almost never want to display literal font tags (and you can use escape characters should you ever need to).

Here is an example using MyFill:

```
MyField "Formatted it looks like ^{MyFill}": 0..1
```

With the inclusion of the curly brackets around the fill, the text looks like this:

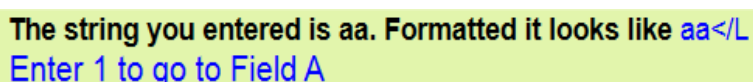


Figure 8. Fills placed inside curly brackets have their font tags interpreted.

Hopefully a future Blaise 5 release will either automatically place all fills in curly brackets, or change the default behavior of fills to interpret tags.

Of course, the fill in Figure 8 above has another issue – the closing tag is cut off and therefore not interpreted.

Fill string size

A complicating factor is that, as in Blaise 4, font tags count towards the declared string length. However, Blaise 5 tags use considerably more characters (e.g. `</F>` vs `@F`, `<newspace>` vs `@\`, etc). This renders many fill declarations insufficient, and we will need to go through all the fills and increase their string length. For example, a string **MyFill** declared as `string[6]` in Blaise 4 will contain 9 characters when the code is converted to Blaise 5.

```
MyFill : string[6]
...
MyFill := '<L>aa</L>'
```

This will result in the string MyFill being truncated and assigned a value of “`</L>`”, as seen in Figure 8. So, the length of MyFill will need to be increased in order to fit the new tag format.

In addition to the incorrect values being assigned or displayed due to this truncation, the issue will have further implications for possible input data and output data fields that may also need to accommodate the increased field length.

It is our hope that this, too, can in the future be handled by the conversion utility. Making such fixes by hand in an instrument consisting of dozens of blocks would be time-consuming.

Keyboard navigation

Unlike in Blaise 4, “out of the box” default layouts in Blaise 5 are geared toward a keyboard-free environment. However, our field representatives and telephone center operators still use laptops and desktops, even as touch-driven devices are being phased in. During the process of converting the NCVS instrument, we explored some ways in which keyboard functionality can be either replicated or emulated in Blaise 5.

Blaise4Layout

One of the samples provided with the Blaise 5 system is the Blaise4Layout resource database. This resource database includes templates that have the effect of imitating the look and feel of a Blaise 4 instrument. Here is an NCVS screen rendered using this resource database:

Figure 9. Using a Blaise4-like layout under Blaise 5.

Note the top control strip that includes some navigation and language selection buttons, as well as buttons for accessing parallel tabs.

This resource database employs a combination of control templates that allow for a keyboard entry of category values (see figure 10). So, Response Value Templates “SetTextBox” and “EnumerationTextBox” allow for keyboard entry of category numbers, while Category Templates “SetCodeText” and “EnumerationCodeText” display the answer category numbers next to the category text.

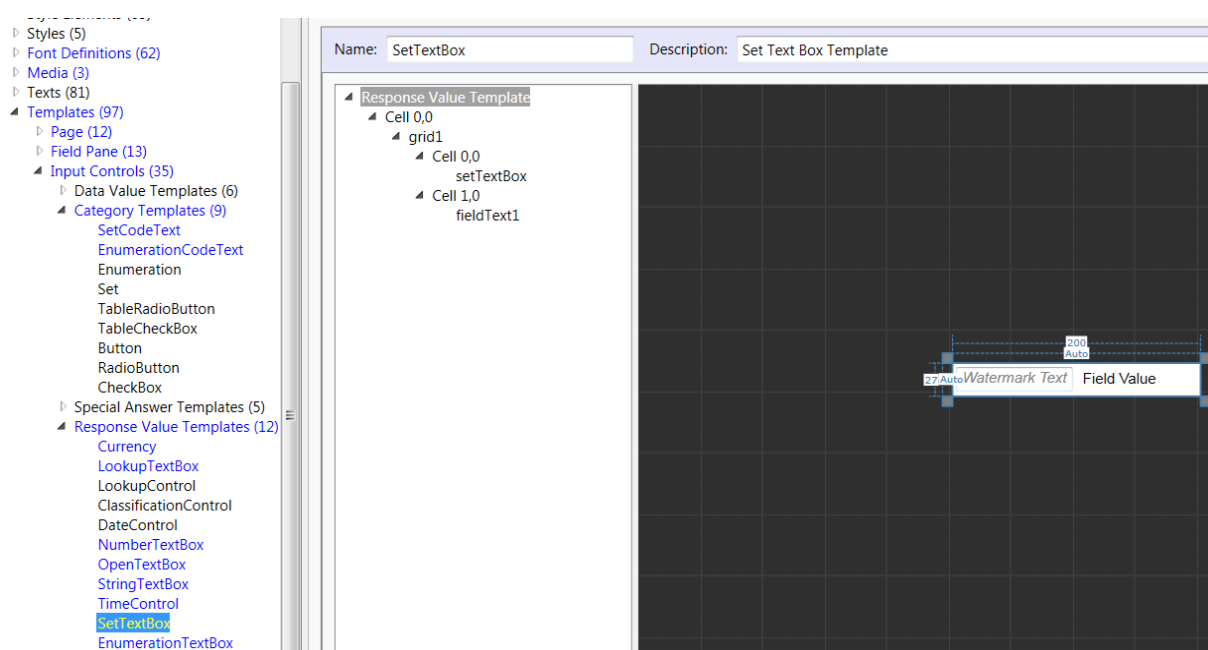


Figure 10. SetTextBox control

Blaise4Layout and its controls are a promising start for integrating keyboard navigation into Blaise 5 surveys, even though it has not yet been fully implemented.

Keyboard shortcuts

Other features of the Blaise 4 DEP that are still missing are the keyboard shortcuts and menu actions. The need for a menu is somewhat obviated by the ability to associate events with buttons. For example, the “Home”, “End”, “PageUp”, and “Page Down” actions can all be associated to button events. But keyboard shortcuts for these buttons are still lacking.

Especially missed is the ability to enter Don’t Know (Ctrl+D) and Refuse (Ctrl+R) responses with a keyboard, and the ability to suppress a signal and call help (F1).

In addition, radio button categories, once checked, cannot be emptied without keyboard input.

Checks and Signals

The handling of checks and signals has changed considerably in Blaise 5. In this section, we will describe some of the changes we have encountered.

Checks involving more than one field / Involving

One of the most useful features of Blaise edit checks has been the “Involving” feature, enabling the user to jump to any of the fields involved in a check in order to correct problem entries. The implementation of the INVOLVING feature in Blaise 5 is lacking. In the following screenshot, fields *Start* and *IncSAM* are “involved” in a CHECK.

The screenshot shows a survey form with four main sections: 'Start', 'IncSAM', 'Mode', and 'BlankRoster'. Each section has a text input field, a 'Suppress' button, and two radio button options: 'Don't know' and 'Rather not answer'. Below the 'Start' and 'IncSAM' sections, there are red error messages: 'If INCSAM = 1 then Start must be 1' and 'If INCSAM = 1 then Start must be 1' respectively. The 'Mode' section has a '1' in its input field, and the 'BlankRoster' section has a '2' in its input field.

Figure 11. Multiple Involved fields on one page

As the screenshot shows, this results in the error appearing next to each of the involved fields. This is okay when all of the involved fields are laid out on one page, but presents problems when involved fields are on separate pages. This is because the interviewer does not know to navigate to the previous pages to correct the error. Even if the field names were included in the error text, there is no simple way to select the involved field and jump to it. Correcting an inconsistency between multiple fields is therefore quite difficult in the current implementation of Blaise 5.

ERROR keyword

An “ERROR” keyword can be used to trigger error checks – both soft and hard – in Blaise 4. It is impossible to know just by looking at this line of Blaise 4 code whether the resultant check will be Hard or Soft:

```
ERROR "@FPlease correct your entry@F"
```

The type of check that will be displayed is determined by the presence of either the SIGNAL or CHECK keyword somewhere in the code prior to the appearance of the ERROR keyword. SIGNAL/CHECK act as a kind of toggle.

We found that in Blaise 5 the behavior of the ERROR keyword was not consistent with Blaise 4 when the SIGNAL/CHECK toggle was not near the ERROR keyword – soft errors in Blaise 4 appeared as hard errors in Blaise 5 in a number of places. Perhaps this is caused by differences in the order in which the Blaise 5 parser reads through the blocks and encounters the toggles. This type of problem can be difficult to notice during testing. It may be good advice not to use the ERROR keyword to trigger checks, unless it is immediately preceded by the toggle, as in “**SIGNAL ERROR...**”

Piping and EditType

Often it is necessary to capture the “Suppressed” value of a signal. In our survey, a suppressed value of a check at the screen ROSTERREVIEW triggers the emptying of the ROSTERREVIEW screen:

```
SIGNAL CK_HHR | (hhr_error = 0)
  if CK_HHR = Suppressed then
    ROSTERREVIEW := empty
  endif
```


In this code, CK_HHR is an EDITTYPE that, via piping, receives the value “Suppressed”, and we then use this value in an *if* statement. The problem is that, in Blaise 5, this syntax gets us stuck in a loop, whereas the value of ROSTERREVIEW keeps getting emptied. A simple modification explicitly empties CK_HHR, and resolves the issue:

```
SIGNAL CK_HHR | (hhr_error = 0)
  if CK_HHR = Suppressed then
    ROSTERREVIEW := empty
    CK_HHR := empty
  endif
```

EditType error text

Additionally, in the same piece of code, *edittype* field CK_HHR contains the error text that, in Blaise 4, displays as the error message:

```
CK_HHR "Please correct this error": EDITTYPE
```

However, in Blaise 5, only the literal condition is displayed, and the text in the EDITTYPE field is ignored:

The screenshot shows the NCVS Questions ver 37.15 (10/27/2014) interface. It includes a table with columns: IN, NAME, REL, AGE, SEX, MARITAL, STATUS. Below the table, there is a section for 'hhr_error = 0' with radio buttons for 'Yes' and 'No'. A blue arrow points to the 'hhr_error = 0' text. The 'Yes' radio button is selected. A 'Suppress' button is visible on the right.

Figure 12. Error Text is not appearing correctly

As a result, we need to define the text inline, as follows:

```
SIGNAL CK_HHR | (hhr_error = 0) "MY TEXT IS DEFINED HERE"
```

Suppressed Checks

There is currently a bug where the switching of tabs “re-awakens” all of the previously suppressed checks. This is perhaps related to the following Blaise language modification:

When a field involved in a suppressed signal is changed by the rules after the signal was triggered, the suppress is removed.

Parallel Tabs

Tab Text

When first converted, the parallel_tab text used block names rather than descriptive titles:

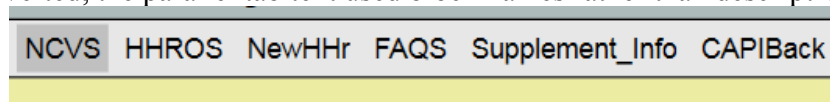


Figure 13. Parallel tab text not displaying correctly

In Blaise 4, this text is set in the Datamodel Properties > Short Text.

In Blaise 5, this needed to be done in 2 steps:

- 1) Specify the text in the FIELDS section of the source code:

```
HHROS    / "HH Roster"      : BlkHHROS
```
- 2) Then, in the Resource Editor > Master Page template > Parallel tabs control > TextSource = TextOrName.
- 3) (Alternatively, if the page template uses Parallel Templates, then go to Page Area > Parallel Templates > [Default] > Parallel Button > Text Source > TextOrName

This is the result:

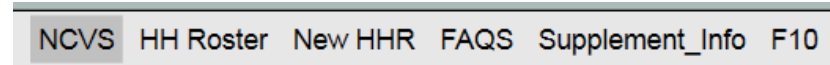


Figure 14. Corrected parallel tab text

Tab visibility

Some parallel tabs should only be visible when a certain condition is met. In our survey, we only want to show the tab “EditBack” when flag WEBEC = 1. In Blaise 4, this is accomplished with the following code:

```
IF WEBEC = 1 THEN
    EditBack
ELSE
    EditBack.KEEP
ENDIF
```

In Blaise 5, the KEEP was causing the EditBack tab to always remain visible, and the statement needed to be commented out. If the KEEP is necessary to preserve the data in that block, a different workaround will be needed, perhaps via the “Toggle Visibility” event.

Navigating at End of Parallel

There is currently no way to specify instrument behavior at the end of a parallel block. One implication of this is discussed below in the section “Exiting the DEP”.

Tables and groups

One of the most important concepts in Blaise 4 is a “TABLE”. This is going away in Blaise 5 and is being replaced by a GROUP construct. Fortunately, this change does not require very big changes to the instrument code.

The GROUP concept is described very well in the Blaise 5 Help documentation. In essence, the block defining the table rows remains unchanged, but the “shell” of the table – the code controlling the routing of the rows is no longer called a TABLE. It is renamed as a BLOCK, and a section called GROUP is added to this block. The GROUP section contains the code (typically a *for...do* loop) for displaying the table rows. The group name is then used in the RULES section to place the entire table on route.

The most difficult part of the process is applying the table template to the group, and modifying that template to accommodate the data types that are used in the table. Additional work is needed in this area to create an optimal layout for large tables with multiple data types, long question texts, and vertical as well as horizontal scrolling. Below is one implementation:

The screenshot shows the National Crime Victimization Survey (NCVS) interface. The title bar reads "National Crime Victimization Survey -- NCVS Questions ver 37.15 (10/27/2014)". The main window has a header with navigation buttons: "To Start", "Back", "Next", "To End", "Language", "NCVS", "HH Roster", "FAQs", "Supplement_Info", "F10". Below the header, there is a yellow box with instructions: "What is the name of the person/people that is/are new to the household?" and "Enter first name on this screen." and "To change a non-member already listed on this roster to a member use the up/down arrow to go to MEMBERCHANGES and enter the reason why this person is". Below the yellow box, there is a table with columns: MEMBERCHANGES, HHROSTER_FNAME, HHROSTER_LNAME, SEX, RELATIONSHIP, HHMEMBER, HSEMUMURE. The table contains one row with data: "Returned from school or college", "David", "Citizen", "Male", "11 Husband", "Yes". Below the table, there is a section for "Demographics BPERSON(1)" with radio buttons for "Person died", "Left for school or college", "Entered institution", "Left because of marriage/separation/divorce", "Person left household for reasons other than above", "Visitor - residence elsewhere", and "Returned from school or college".

Figure 15. Attempt at a Blaise4-style table in Blaise 5.

Critical Fields / Re-execution

Critical fields vs. Server contact option

The layout view contains an option to “Generate Critical Fields”. This option requests server contact as soon as a value is entered into the “critical field” allowing for the refreshing of values and routing prior to leaving the page. This option is important for web surveys, but it is not relevant for stand-alone CAPI survey applications.

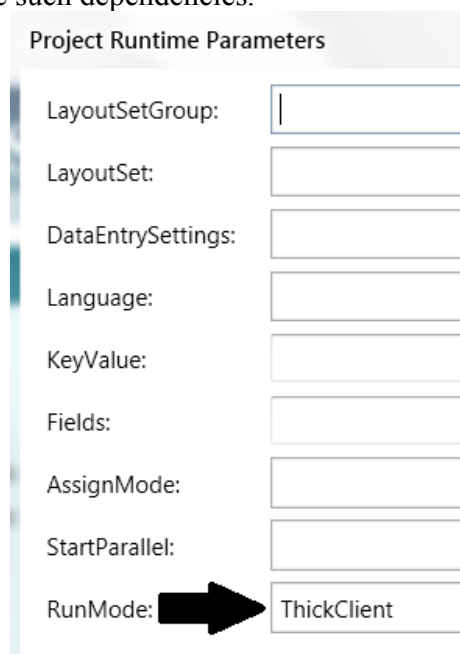
Instead, a project-level setting must be changed in the project’s .blax file: select the **Settings tab > Advanced > Server Contact > check “After leaving a changed question”**.

Run Time Performance Delay

When first running the instrument, we noticed a significant re-execution delay in Blaise 5 for our large survey, with performance being inferior to the Blaise 4 instrument running side-by-side on the same machine.

Specifically, there was no delay whatsoever in navigating between most fields in Blaise 4. In Blaise 5, when running the NCVS datamodel, there was a delay of 1.5 to 2.5 seconds in updating a fill, or in changing which field comes on route based on a value entered on the same page.

This issue was improved by passing a “ThickClient” parameter at runtime. The delay went down to about half a second for fields without any rule dependencies outside their local block. The delay is longer when there are such dependencies.



Project Runtime Parameters	
LayoutSetGroup:	
LayoutSet:	
DataEntrySettings:	
Language:	
KeyValue:	
Fields:	
AssignMode:	
StartParallel:	
RunMode:	ThickClient

Figure 16. Runtime parameter *ThickClient* speeds up performance.

This was a substantial improvement, but it still does not yield the type of instantaneous smooth flow that one sees in a Blaise 4 instrument. More than just a delay in going from one screen to

another is first landing on a field that will then be taken off route when the rules engine catches up half a second later. This has the effect of being occasionally disorienting, as the active field suddenly switches or a check fires for a field that has already been passed. If the question text consists of a fill that is half-a-second late in recalculating this means the interviewer will begin reading the wrong text that will then change mid-sentence.

In summary, we believe there may be two separate issues involved. Firstly, there is a greater processing delay than in Blaise 4. Secondly, unlike in Blaise 4, the application allows the focus to progress to the next field even as the prior field is still being evaluated in the background, resulting in disorienting behavior.

Data storage and saving

Data and metadata from the open, incomplete, data entry session is continuously being saved by the Data Entry Service into a database called *RuntimeSessionData2.db*. This includes not only the values of all the fields, but also metadata such as the active field and so on. This database is not the same as the permanent instrument database. Field values are moved into the instrument database only after the last field on route in the form is reached – or via a Save action.

This session data is used to back up a partial case, and enables the user to re-enter the case in the same state in which it was last left.

If the session is terminated before the end of the survey is reached, the permanent database will not be updated.

One of our key business requirements is the ability to retrieve data from a partial case in the field. During our Blaise 4 field operations, we often handle transmissions of partial cases for various purposes, and being able to easily load these partial cases between laptops, or to view them is very important. Sessions sometimes crash mid-interview, or a laptop may experience a power failure. Sometimes a partial case may need to be re-assigned to a different interviewer. We are not sure whether it is possible to extract and copy data for an individual case from the session database, but if it is it will require a third program accessing it through the API.

In order to actually store the data from a partial case into a permanent case database, an explicit *Save* action needs to be added to instrument controls. For example, one can add the *Save* action to the OnClick event of “Next” or “Back” buttons, so that the data is saved at each page switch:

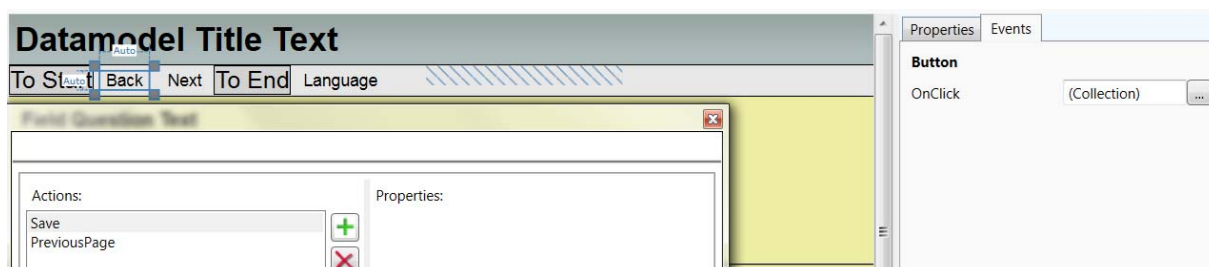


Figure 17. Adding an event to buttons

Similarly, the Save action can be added to parallel tabs, all other buttons, radiobuttons and checkboxes. It seems to us that this would be good practice, though of course it would have to be weighed against potential performance costs. It must be said that unlike in Blaise 4, I did not notice any delay in Blaise 5 associated with the Save action. It is not possible to add events to text boxes associated with integer and string fields; it is not possible to request that data be “auto-saved” based on a set time interval (a feature that we utilize in Blaise 4); and it is not possible to associate a Save event with the closing of the data entry program.

The new runtime session database with a more robust implementation of the Save action promises to be a good way to deal with partial and abruptly terminated interviews.

Exiting the DEP

Perhaps the most important functionality in this regard is preventing interviewers from exiting the instrument without first going through the “Back” parallel block in which they gracefully wrap up the interview and at the end of which the data is saved and the form is closed. It is currently not possible to disable the Windows “X” button on the DEP form in order to prevent the user from immediately exiting the survey session. This is perhaps due to the multi-platform target of Blaise 5 – different platforms have different rules for shutting down applications. Still, controlling how an application is shut down is an important element of the CAPI survey environment. It would be very useful to have the ability to disable the X, or to trap and modify the “close” event of the application.

Control Center and DEP environments

The Control Center environment is a big step forward from Blaise 4. Real-time parsing, Intellisense editor, device emulator, and the WISYWIG layout designer are welcome improvements.

The Resource Editor is a complex tool that was at first intimidating and cumbersome to use. However, once an understanding of how templates work together was attained, the editor became much easier and more intuitive to use. Thus, appropriate training is essential.

The watch window and database browser are not yet fully implemented.

Speed

The speed of the control center and layout designer has improved dramatically since earlier iterations of the Blaise 5 software. It is now practically seamless. Certain particularly intensive operations (such as generation and removal of critical fields throughout the whole instrument) still take a very long time, but on the whole the control center operations are quite smooth.

One notable issue is the solution build. When new pages need to be generated, it takes 10 to 12 minutes for our survey to generate all of the pages. By contrast, the same survey in Blaise 4 took about 10 seconds to prepare on the same machine.

```

15:34:13 : Generating layout
15:34:13 : WARNING: Not all field references of the
15:34:13 : Layout generated
15:34:13 : Generating meta file
15:34:15 : Generating interview pages
15:35:27 : Generated page 100
15:35:39 : Generated page 200
15:35:58 : Generated page 300
15:36:19 : Generated page 400
15:36:41 : Generated page 500
15:37:06 : Generated page 600
15:37:29 : Generated page 700
15:37:50 : Generated page 800
15:38:13 : Generated page 900
15:38:36 : Generated page 1000
15:38:57 : Generated page 1100
15:39:19 : Generated page 1200
15:39:40 : Generated page 1300
15:40:01 : Generated page 1400
15:40:22 : Generated page 1500
15:40:44 : Generated page 1600
15:41:07 : Generated page 1700
15:41:33 : Generated page 1800
15:42:00 : Generated page 1900
15:42:26 : Generated page 2000
15:42:52 : Generated page 2100
15:43:17 : Generated page 2200
15:43:43 : Generated page 2300
15:44:10 : Generated page 2400
15:44:36 : Generated page 2500
15:45:02 : Generated page 2600
15:45:24 : Generated page 2700
15:45:47 : Generated page 2800
15:46:11 : Generated page 2900
15:46:26 : Generated page 3000
15:46:27 : Generated 3011 pages
15:46:27 : Generating datamodel definition
15:46:27 : Generating check definitions
15:46:27 : Generating classification types
15:46:27 : Generating custom pages
15:46:29 : Meta file generated
15:46:29 : Build successfully completed (217 warnings

```

Figure 18. Generating pages during a Build takes 12 minutes.

There is an important option in the control center to “*Build without generating pages*” -- this means that pages will be generated at runtime. You can speed up the build process even more by un-checking the option to *GenerateAllSections* in the “Properties” menu of your blax’s Layout view. (However, if you have layouts that apply to specific sections, such as blocks or groups, then you need to set this option to “True” in order to generate accurate section indexes for your Layout tasks.) It seems that if you are not working on section layouts you can uncheck this option.

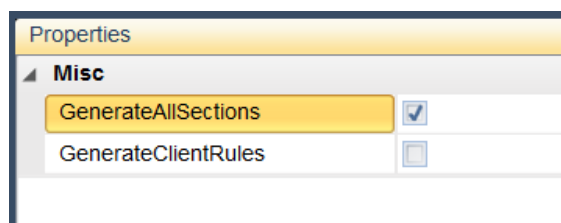


Figure 19. This option generates Section Indexes.

Blaise documentation warns of a run-time performance cost to this build option. We have not noticed the cost penalty thus far in our CAPI standalone environment, and it certainly makes the build process much more manageable. If we do start noticing a cost penalty, then we may decide to use the option during development only, but do a full build with pages when preparing for a release.

System Errors

During our testing we sporadically encountered the following errors in the DEP. It is not clear what exactly triggers these errors. More information about the source of the errors – and perhaps a facility to report the errors and the call stack to the developers would be helpful.

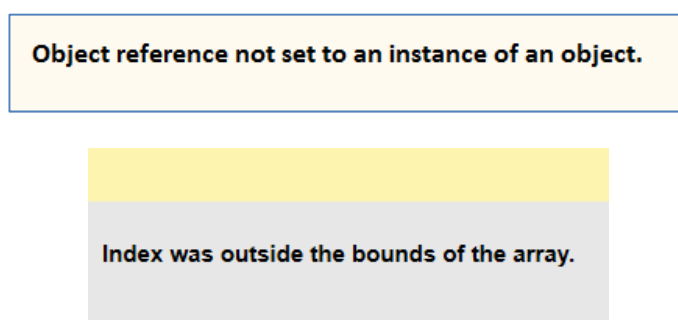


Figure 20. DEP system errors

Additionally, when a data entry session is left inactive for too long – about 30 minutes? -- the following error screen appears at the next page switch, and the session is automatically terminated:

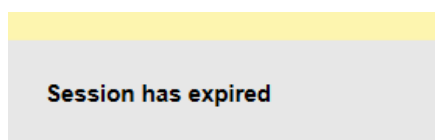


Figure 21. "Session has expired" error message in the DEP

Terminating a session due to inactivity is a questionable way to end an interview session and can potentially be a source of interviewer frustration and data loss. If there are environments under which a session expiration limit is preferred (such as due to a security requirement in a web

survey), then perhaps there should be a survey-level option to disable the expiration timer in other environments, or at least to save the data in the permanent database before terminating the session.

Item-level Help screens

Item-level Help screens were accessed in Blaise 4 via an F1 function key. In Blaise 5, item-level help can be accessed by defining a Help Role for a field, and attaching an event to a Help button.

First one needs to define a “Help” role for the datamodel, and map it to the Help Role defined in the Layout Designer:

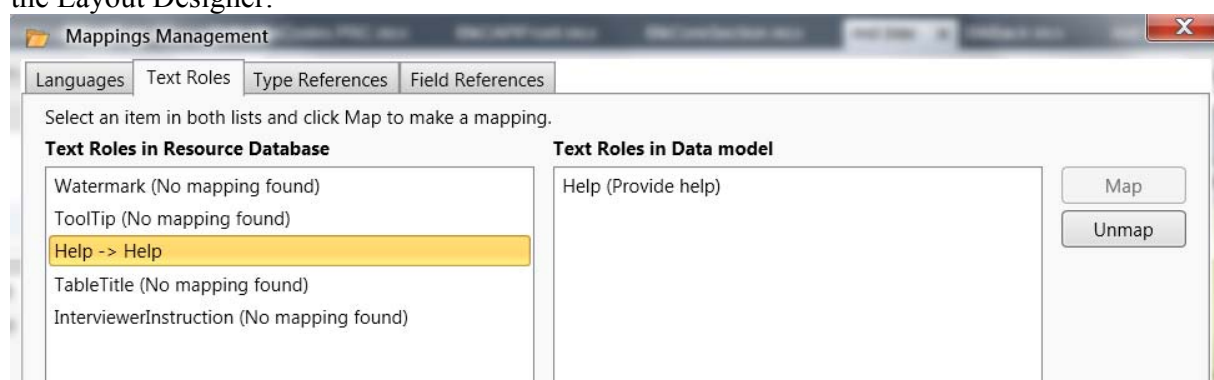


Figure 22. Mapping the Help Role

Then, the HELP role tag is defined for fields which require item-level help:

```
START_CP ( "START_CP" )
  ENG "<newline><L>  Some question text...</L>"
  HELP "Help text here..." / "Start CAPI Interview" : TStart_CP, NODK,
NORF, NOEMPTY
```

Lastly, an event is attached to the “Help button” in the Resource Database. That event can be a *GoToURI* , such as file:///C:/NCVS/Blaise5/NCVSver37_15/FunctionKeys.htm

The Help Button has a regular expression associated with its Visibility property – it is only visible on fields for which a Help Role is defined:

```
IF LEN(Field.GetRoleText('Help')) = 0 THEN 'Collapsed' ELSE 'Visible' ENDIF
```

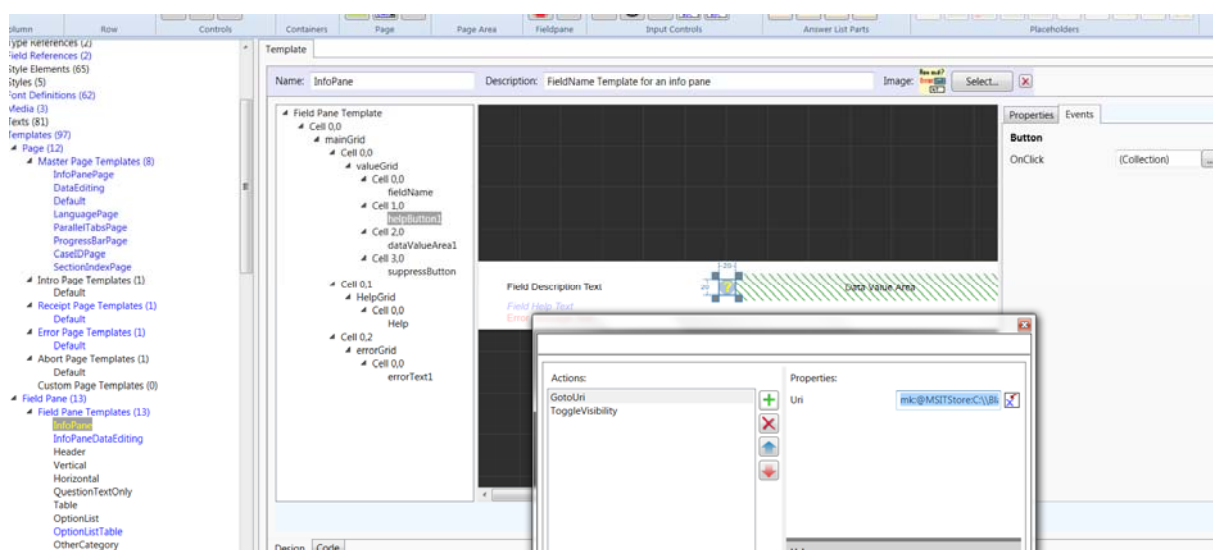


Figure 23. GoToURI action

To access a page in compiled HTML help (.CHM), one can use the following syntax:

```
mk:@MSITStore:C:\Blaise4
surveys\ACS\m201501\help\ACS_WITH_TOC.chm:/H_WKL_PRS.htm
```

This will bring open a new window with the HTML Help screen.

Now we can create a Regular Expression so that item-specific help pages can be brought up. In our case, help pages are named in a standardized format *H_<field name>.htm*. Thus, the regular expression can look like this:

```
'mk:@MSITStore:C:\NCVS\m201501\help\NCVS_WITH_TOC.chm:/H_' +
Field.LocalName + '.htm'
```

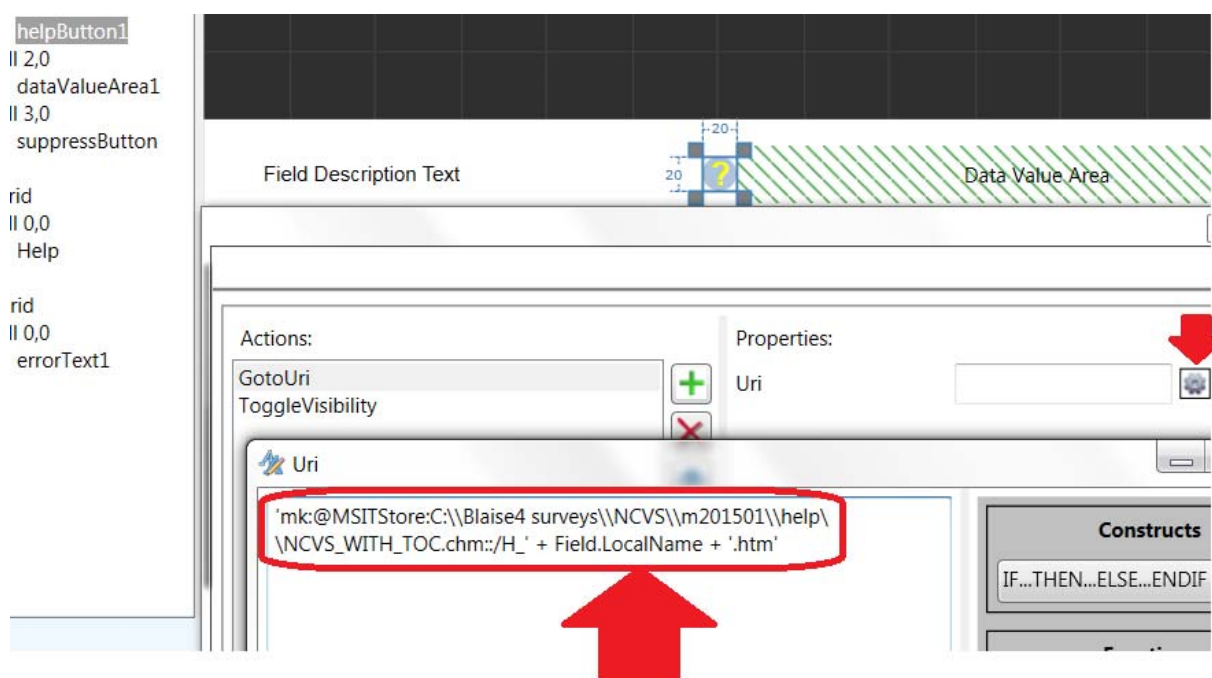


Figure 24. Expression to create dynamic item-level help based on the name of the current field.

As a result, the help screen opens in a new browser window, as illustrated below.

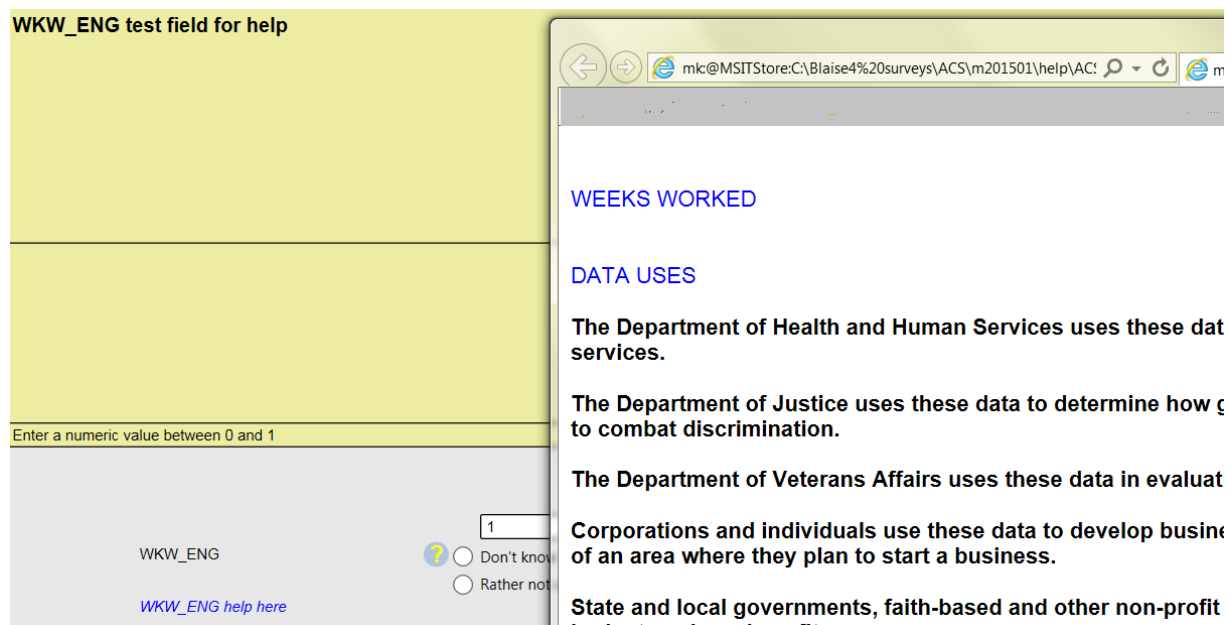


Figure 25. Item-specific help displayed in a browser.

Data Values

NCVS uses a number of storage fields that are never asked or displayed, but that retain important calculated data that is used either to route fields during the interview, or as output. During our testing we keyed identical complete cases in Blaise 4 and Blaise 5, and then compared values of 20 such fields. All of the values matched. In addition, we tested each of the possible outcome codes (i.e. codes for a complete interview, partial interview, sufficient partial interview, refusal, no such address, etc.), and in all cases Blaise 5 data matched Blaise 4 – all of the outcome codes were achieved.

Audit trail access

Audit trail data is stored in a SQLite database called *AuditTrailData.db* – a database shared across all installed surveys. This is different from Blaise 4, where audit trails are stored in a plain text file, and a separate audit trail file is associated with its own case. As such, one cannot view the audit trail without an SQLite browser, and even inside the SQLite browser it is difficult to inspect the content of the audit trail.

Here is a screenshot of the audit trail data inside a third-party SQLite browser:

	SessionId	InstrumentId	TimeStamp	Content
	Filter	Filter	Filter	Filter
1	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<StartSessionEvent Width="1536" Height="864" Device="WindowsDesktop" Language="ENG" Platform="Windows" />
2	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<UpdatePageEvent LayoutSetName="Interviewing1" ParallelName="PRIMARY" PageIndex="1" />
3	{662c1384 f...	{bf4b0c63 ...	2014 12 24 1...	<EnterFieldEvent FieldName="CaseID" AnswerStatus="Empty" />
4	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<KeyboardEvent KeyStrokes="00000004[BACK]1" />
5	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<LeaveFieldEvent FieldName="CaseID" Value="00000001" AnswerStatus="Response" />
6	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<ActionEvent Action="Save()" />
7	{662c1384-f...	{bf4b0c63-...	2014-12-24 1...	<EndQuestionnaireEvent />

Figure 26. Content of *AuditTrailData.db* *EventData* table

This view includes all sessions from all cases from all installed instruments. A separate API utility will most likely need to be written to enable case-level access to audit trails.

Effort

Initial conversion of a large instrument from Blaise 4 to Blaise 5 involves significant effort. This effort will certainly be much reduced over time, as the organization settles on standard approaches to Blaise 5 layout, deployment, data storage and transmission; as programmers become more familiar with working on Blaise 5; and as Blaise 5 itself matures.

Still, thorough testing of all aspects of instrument functionality is required, and one should not assume that the converted instrument will perform as it did under Blaise 4.

At present time, we have been able to convert into Blaise 5 much – but not all – of the functionality of our Blaise 4 NCVS instrument. As mentioned earlier, emptying of blocks is a key feature for NCVS and it is not yet implemented. This limitation, together with the lack of Manipula implementation, keeps the Blaise 5 NCVS instrument from being fully functional.

In addition, we still need to find the optimal layout approach that takes advantage of the new touch-driven technology while also retaining the functionality achieved in Blaise 4 via keyboard shortcuts and menus. We also need to better understand how the new Blaise data storage approach can work with our data transmission processes.

We still have a ways to go in order to put this all together.

Summary

Blaise 5 promises to be a robust data collection system. It builds upon the capabilities of Blaise 4 with a much enhanced, modern Control Center, multi-platform capabilities, and a complex layout design capabilities.

Perhaps the most significant area of concern is the performance speed in the DEP. Processing speed will need to improve in order to smoothly run large household surveys under Blaise 5.

Key positives and negatives we encountered during this conversion exercise are summarized in the table below.

Features that convert well	Features that need work
Overall Blaise Rules performance	CAPi DEP Performance speed
Help	Graceful exit from a case
Parallel tabs	Watch Window
Control Center environment	Viewing Audit Trails
Tables / Groups	Handling of checks
Fonts	Keyboard-driven navigation / DK / RF

Table 1. Summary of impressions

This paper focused primarily on replicating Blaise 4 functionality under Blaise 5, and we noted a number of features that may be lacking at this stage in development. Left unexplored are many new features and capabilities of Blaise 5 that may offer entirely new possibilities for future survey design.

The views expressed in this paper are those of the authors and not necessarily those of the U.S. Census Bureau.

Developing the UK International Passenger Survey in Blaise 5 on Tablet Computer

Lanthao Benedikt, Office for National Statistics, United Kingdom

1. Introduction

Over the last years, stringent budget cuts have placed tremendous emphasis on the need for ONS to modernise our data collection process in order to continue producing high quality and timely outputs, while coping with increasing demands on the complexity of the information to collect.

In this paper, we report on our latest progress towards transferring the International Passenger Survey (IPS) from paper form to electronic questionnaire on tablet. The focus will be placed on design concepts and usability assessment of the new survey instrument. This is also the opportunity to showcase the many possibilities offered in Blaise 5, not only in terms of routing and validation checks, but also – and above all - in terms of questionnaire layout.

2. What benefits for the survey?

The IPS is a continuous survey that has been conducted by ONS since 1961. Between 700,000 and 800,000 interviews are collected every year at all major airports, seaports and tunnel routes to and from the UK. This is the main source of information used by the UK governing bodies for planning, monitoring and informing decisions on tourism and immigration policies.

At the moment, the IPS data are collected on paper forms which are then manually keyed into the CADI system in Blaise 4.8. The data are then transferred electronically to ONS headquarter where further cleansing and validation take place.



It has been a goal for ONS for some time now to computerise the IPS data collection. However, until recently, we felt that the hardware was not ready [1]. With recent progress made on hand-held devices, as well as the availability of Blaise 5 on Windows tablets and the prospect of using the software on cheaper Android tablets in a near future, ONS has started exploring this option again. Our vision is three-fold.

1. Speed up the data collection and processing time thanks to the elimination of the CADI task. Take advantage of the in-built data validation checks and logical routing capability in Blaise to cut down on human errors, hence improve data quality.
2. As a direct consequence of the previous objective, assess whether this innovation offers a cost-effective solution that allows ONS to make savings.
3. Last but not least, exploit the layout capability in Blaise 5 to design a survey instrument that is fit-for-purpose, more user-friendly to interviewers and more engaging to respondents. This will improve the image of ONS as a modern organisation that embraces changes and stays on top of the latest technologies.

3. First prototype of the Blaise 5 questionnaire on tablet computer

The project started in July 2014 with an initiative of the IPS research team in Newport who organised a visit to Bristol airport. A Blaise developer was invited along to shadow IPS interviewers for a day in order to understand the nature of field work and to investigate where an electronic questionnaire on tablet could add value. Three main areas for improvement were identified.

1. Survey flow: the IPS questionnaire has a complex routing with a Main Questionnaire and various trailers e.g. Immigration Trailer, Employee Trailer, Student Trailer short/long forms [2][3], making it sometimes difficult to quickly identify the next question to ask, especially for inexperienced interviewers. There are potentials for human errors and possible negative impact on response rate.
2. Coding frames: IPS interviewers usually carry a folder with all the lookup tables that they need for filling out the survey. Examples include flight numbers, airport IATA codes, country codes, region codes, UK towns, to name only a few. Other documents they carry include the trailers mentioned above, as well as “show-cards” in 16 different languages. It is quite difficult to shuffle through the folder to quickly find the correct forms, even more so because IPS data are collected on the move. During busy shifts, interviewers often need to navigate through crowded areas while conducting the survey.
3. Data keying (CADI): this process is unanimously perceived as very time consuming and cumbersome. So much so that most field managers have to take work home to complete their tasks on time. Manual data keying also has potential to introduce many errors, the elimination of this step is probably where the most significant improvement can be achieved, both in terms of efficiency and data quality.

The good news is that most of the problems mentioned above can be solved with Blaise routing capability, in-built checks, lookup features. Thus, a first prototype of the electronic questionnaire was very quickly developed and deployed on a test tablet computer - a Lenovo Miix2 8-inch Windows 8.1 – in order to explore what could be done. Examples of screenshots are shown in Figure 1.



Figure 1- Look-and-feel of the first prototype of the IPS questionnaire in Blaise 5. Different background colours were used to differentiate various trailers and the main questionnaire.

By end of August 2014, the IPS research team organised a focus group meeting in Newport during which the prototype was demonstrated to a selected group of interviewers and field managers from Bristol, Manchester and Liverpool airports. The main aims of this meeting were to gauge users’ first reactions to the new survey instrument and to gain feedback on what could be improved. The results are summarised in Table 1 below.

Positive	
Survey Routing	Perceived as a major benefit in moving from paper to tablet
Lookup / Coding frames	Perceived as a major benefit for new interviewers. However, for seasoned IPS interviewers who know by heart most of the coding frames, it would be faster to type the code than use

	the lookup
General look-and-feel	Perceived as engaging and user-friendly
Number of questions per page	Good choice of number of questions to display per page. Questions that are logically related are displayed on the same page. Good use of conditional visibility (e.g. show free text field only if user clicks on "Other", hide it otherwise)
Fonts and colours	Good choices of fonts and colours. Interviewers like the use of different background colours to differentiate main questionnaire and various trailers
Edit checks	Are very helpful but to be used with moderation. Too many checks can be frustrating and cause respondents' drop out. Should allow skipping of "soft errors"
Need improvements	
Screen space usage	Need to maximise space allocated for displaying the questions. Interviewers unanimously suggested that the title panel be removed to save more space for displaying the questions
Lookup	Reduce number of items shown in lookup shortlist. Currently showing too many items which hides part or most of the screen, including the question
Use of radio buttons	It may be difficult to select the radio buttons and some of the answer options
Minimise user action	Make the best use of Blaise to implement as much as possible what can be automated e.g. time of interview, currency conversions, summing and checking expenditure etc. so interviewers will have less to fill in
Other suggestions/requirements	
Introduction page	Add an informative first page that looks official with ONS logo that interviewers can use to introduce themselves and the survey to respondents
Streamline paper questionnaire	There exists a number of issues with the current paper questionnaire in relation to question wording and routing. This is a good opportunity to streamline the survey questions
Coding frames	A number of inconsistencies have been found in the current coding frames during the implementation of the electronic questionnaire. A review and cleansing of these databases will improve the performance of the lookup search
Edit checks/Key questions	A review of the current edit checks by the IPS research team is necessary in order to specify hard checks and soft checks. The interviewer can ignore soft errors and skip non-key questions. A mechanism needs to be agreed on how to deal with soft errors at a later stage
Routing/Formatting	When transferring the questionnaire from paper to tablet format, some formatting and question ordering need to be modified. Further discussions with the IPS research team will be needed to ensure that such modifications will not introduce any bias to the survey
Use of Date picker	As a real life example, to the question "When did you last arrive in the UK?", respondents often do not answer with a precise date, but rather something along the line of "2 weeks ago", "last July" etc. On the paper form as well as in the first electronic questionnaire prototype, date fields are presented as a text box to fill in dd/mm/yyyy. Interviewers prefer the use of a date picker that would be more helpful to identify the correct date.
IPS sample selection	Can the electronic questionnaire be programmed to handle IPS sample selection such that the adequate form will come on screen at the right time? The IPS samples every 10th person, of which only a selected number will be interviewed on migration
Survey flow/ Navigation	To speed up the interview process, it is preferable to set the survey to automatically advance to the next field/page as soon as the current question has been answered, without the interviewer having to click on the "Next" button. (Request added subsequently after tests carried out with field managers in Titchfield)

Table 1 - Summary of main findings from the first focus group meeting with interviewers and field managers from Bristol, Liverpool and Manchester airports in August 2014.

4. Findings from literature review

In addition to the research undertaken with the IPS focus group, a literature review was also carried out, which aimed to gather useful information on questionnaire design for mobile devices and to gain awareness on the major pitfalls specific to conducting Passenger Surveys.

4.1 Questionnaire design best practices

Although various guidelines exist for best practices in design matters, one of the most exhaustive studies is no doubt “The Design of Every Day Things” by Don Norman [4], who is widely considered an authority in the fields of artificial intelligence, usability engineering and human perception. In a nutshell, a subset of the standard usability principles outlined in his book that are relevant for questionnaire design are summarised below.

- Visibility: use large Sans Serif Fonts that look crisper on a computer screen (e.g., Arial instead of Times New Roman), use large selection boxes, and contrasting colours. Provided orienting information such as question number and use bold text, underlined text for important information.
- Feedback: give users information about what action has just happened, for example boxes change colour when selected, providing users confirmation that the response was registered. Informing users that something is happening during delays e.g. “Loading... please wait”.
- Constraints: Limiting choices can prevent errors, e.g. remove the internet browser buttons so that users would not accidentally use them.
- Mapping: this refers to the relationship of the application controls to their effect in the application. Response options are presented in order of their numeric scale values, lowest on the left to highest on the right. Navigation boxes with Back on the left and Next on the right.
- Consistency: use the same elements in the same way throughout the application, which reduces the need for explanations and enhances memorability. All questions in the survey must be presented consistently with the same font. All navigation buttons are the same colour and appear in the same position on all pages.
- Affordance: provide hints or clues about how the application works. For example, a labelled square on a touch screen “affords” pushing; users recognize that the square works as a button. Using only clickable buttons as input controls makes the survey simpler and more intuitive.

4.2 Lessons learned from related works on conducting Air Passenger Surveys

The Transportation Research Board (TRB) of the US National Academies has invested significant research effort in the field, the result of which is reported in the Airport Cooperative Research Program (ACRP) Guidebook [5]. This is a valuable reference for planning all aspects of Air Passenger surveys, with supporting evidence from two recent examples of implementation at Los Angeles Int’l airport in 2011 [6] and at Phoenix Sky Harbor Int’l Airport in 2012 [7].

Overall, recommendations that can be found in these report are to great extent in agreement with Don Norman’s principles, and in addition, they offer many interesting insights into the practicality of conducting Air passenger Surveys.

These include in particular the reality of having to deal with strict security measures inside airports (e.g. no Internet connection, therefore no online fancy applications such as Googlemaps). Another challenge is the nature of the travelling public. Someone completing a survey questionnaire in the comfort of her home will be more cooperative than someone who is walking through a busy airport terminal and is anxious to catch a flight. Therefore, great care needs to be taken to design a short survey that flows smoothly: making the right decision on navigation ease, routing, lookups, deciding on the optimum number of edit checks etc. are essential.

5. Second prototype of the Blaise 5 questionnaire on tablet computer

Feedback gained from the focus group discussions and results of the literature review shape the development of the second prototype of the Blaise questionnaire as shown in Figure 2. With this new version, the routing remains very much unchanged. The main improvements concern the look-and-feel of the electronic questionnaire and the survey flow (ease of navigation).

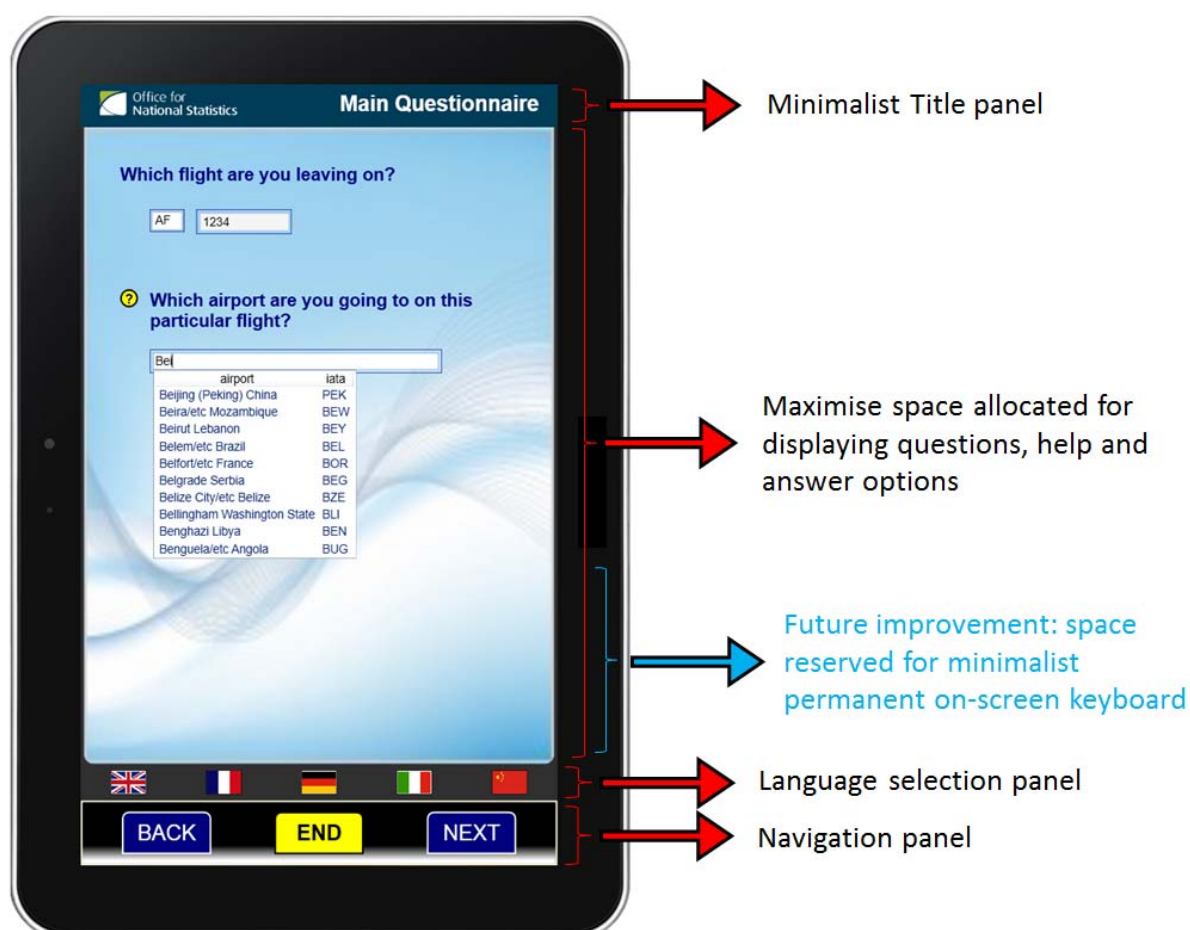


Figure 2 - Look-and-feel of the second prototype of the Blaise questionnaire on tablet

5.1 Page layout

The main improvement in the page layout is the application of Don Norman's principle of *Consistency*. The page template is now designed with fixed sizes such that the various components of the page always appear at the same positions as the user navigates through the survey. This is different from the first prototype where, for example, the navigation buttons BACK and NEXT changed their vertical positions between pages.

Don Norman's principle of *Visibility* is also enforced by the use of contrasting colours and panel borders to better differentiate the various components of the page. For the most important part of the page - the content area where the questions are displayed - a lighter colour is applied to make it stand out more and ensure that the user's attention will be focused in priority on this area than on the rest of the page. Another part of the page that is of importance is the navigation panel. Shading and border are applied to make the panel raise in relief, while other panels that are less often used e.g. the Language panel and the Title panel are designed to look more "retreated".



Figure 3 - Use of different background colours for different parts of the survey

Taking into account feedback from the focus group, the title panel is now made minimalist, with only the ONS logo and the “Page Title” to indicate whether the user is within the Main Questionnaire or within one of the various trailers. Different background colours are also used to indicate various parts of the survey as shown in Figure 4, interviewers previously fed back that they found this useful.

5.2 Answer selection. Font and font size

One major design element was deciding how interviewers would select their responses for the survey questions. A standard radio button's target area for selection was perceived as too small.

Applying Don Norman’s principles of *Visibility* and *Feedback*, we use answer boxes and design these as large as possible. Gradient shading is applied to the boxes to create an “embossed” impression and make these look more like buttons, thus enforce the principle of *Affordance*. A visual cue is added to feedback to the user when a response had been selected; the colour of a selected response button immediately changes to green, as shown in Figure 3.

Large Sans Serif fonts are used for question text, answer options and help text. The same templates are applied across the whole questionnaire, in accordance with the principle of *Consistency*.

- Arial 30 Blue Font for Question text.
- Arial 22 Black Font for answer options.
- Arial 20 Dark Slate Grey Italic Font for Help text.

A question mark is displayed next to each question if Help text are available. Clicking on the question mark will expand the help. Clicking again on the question mark will collapse the text.

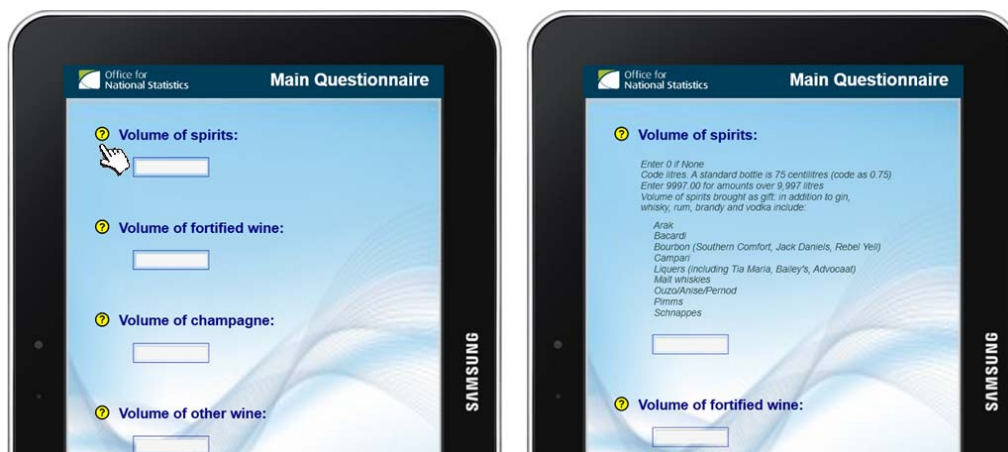


Figure 4 – Help text expand/collapse

For questions where the answer options exhibit a logical order or scale e.g. satisfaction survey questions, we display the answer options in the order lowest value at the bottom of the page to highest value at the top, in agreement with the principle of *Mapping*, as shown in Figure 5.



Figure 5 – Applying Don Norman’s principle of *Mapping*

5.3 Survey flow. Navigation ease

To improve navigation ease, when a question is answered, the survey automatically advances to the next question. This is achieved by setting the ‘On Value Change’ action of the button to ‘Next Field’ in the Blaise 5 Resource Editor. This has been for most questions whenever it is possible e.g. selection buttons, text lookups, number lookups, date pickers, dropdowns etc. However, there exist a number of exceptions where the user has to explicitly click on the NEXT button to advance to the next question:

- Questions where multiple answer options can be selected e.g. “*In which countries have you been living/working/studying for the last 12 months?*” There is no a priori knowledge of how many options the user is going to select so we cannot set the automatic advance to next field.
- Fields using String Text Box or Number Text Box. If we set the ‘On Value Change’ action of the button to ‘Next Field’, as soon as the user types the first letter or number, the cursor jumps to the next field/page, which is not the desired behaviour.

This is a trade-off between *Navigation Ease* and *Consistency* where, in order to improve the survey flow, we choose to implement the automatic advance feature. But we lose in consistency.

5.5 Multiple languages. Lookups. Coding frames

With the second prototype is introduced the multiple language feature, as shown in Figure 6. Clicking on the national flags at the bottom of the page will spontaneously switch the screen display to the desired language.



Figure 6 - Multiple language questionnaire. Examples shown for English, French and Chinese

As discussed earlier in Section 3, the IPS employs a large number of coding frames and multiple “show-cards”. These need to be examined on a case-by-case basis in order to identify the best implementation for each type. For example, a date picker is the most suitable implementation for date fields, while a lookup text box with Trigram search would be the best choice for airport IATA codes, country codes, UK towns, etc. as shown in Figures 7 and 8.

Figure 7 - Lookup text box with Trigram search is the most suitable implementation for fields where the user may be unsure of the correct spelling e.g. airport codes, UK towns, country codes

Figure 8 - A Date picker is used for date fields. In some cases, when the number of options in the lookup list is limited, a simple dropdown is sufficient, as for "Reason for Visit"

A particular type of lookup needs to be designed for question 6 in the IPS questionnaire. If the respondent's country of residence is either Belgium, Canada, Germany, India or Spain, then he/she will be asked to select his/her city of residence from a list of important cities within the corresponding country. If a city cannot be found from that list then ask for the region/state/province of residence.

The IPS research team and IPS field managers have indicated that an implementation using interactive maps would be highly desirable. However, in the absence of Internet connection inside airports, we cannot rely on any off-the-shell solutions such as Googlemaps etc. and we have to design an offline in-house solution, as shown in Figure 9.

Here, an example is shown for Germany. The respondent is presented with the map of Germany, displaying the main regions. After selecting the region

of residence, the respondent is presented with a further list of cities to select from. Compared to the paper form, the order is reversed *tablet:(regions → cities)* instead of *paper:(cities → regions)*. The main reason for this is because there is no sufficient space to display all the cities on the map. Changing the order of the questions should not introduce any bias to the survey in this case since the answer is of deterministic nature.

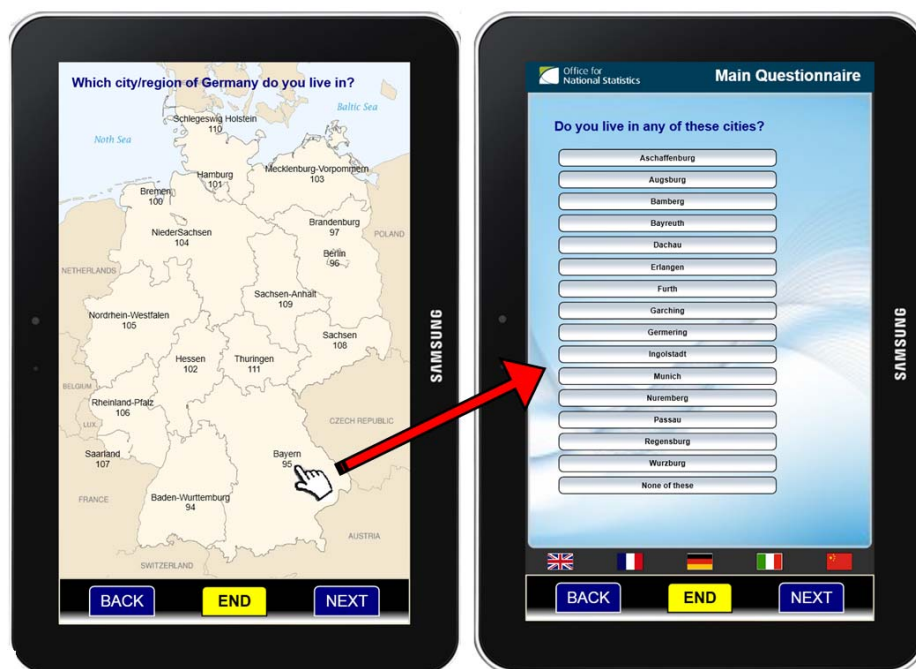


Figure 9 - In the absence of Internet connection, an offline solution was designed to allow the user to select regions/provinces on a map

The implementation of this feature is by far the most cumbersome in the whole project! The country map is applied as the background of the page. The Title panel and the Language Selection panel are removed to save more space for displaying the map. Over the map is superimposed a complex network of grids inside each cell is a selectable button. The button has a transparent background and no border, the button text is the name of the region. The button “On Click” action is a collection of actions: when the user touches on the region name, the region code (e.g. 14 for Antwerp in the example shown in Figure 10 below) is assigned to a field called *RegionCode*. Based on this value, the list of cities for Antwerp will be displayed on the next page for the user to select.

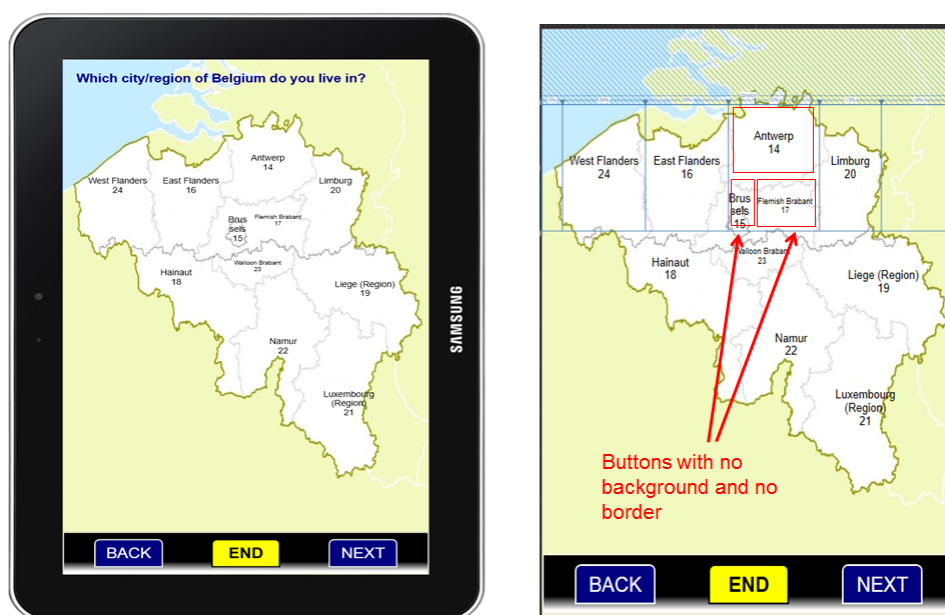


Figure 10 - Map of Belgium with selectable provinces names. On the right hand side: how it is designed in the Blaise 5 Resource Editor

Designing the network of grids for the maps is quite time consuming. This is necessary because in the Blaise 5 Resource Editor, a button can only be placed inside a grid cell. Below is an example shown for the map of Canada. Designing the network of grids for Germany, Belgium and Canada is relatively easy, but for countries with numerous regions, provinces and states, the grid networks can become quickly very intricate, as for India and Spain.



Figure 11 - Designing the network of grids becomes quickly very complex for countries with numerous provinces and regions such as India or Spain

6. Testing

For the time being, it has been decided that the first Usability study would only focus on testing the IPS Air Departure questionnaire at a limited number of airports.

The IPS questionnaires used for seaports and tunnel routes are very similar to the airport questionnaire, the wording is slightly different for Arrival compared to the Departure questionnaires, but not much, so a minimal effort will be needed to adapt the Air Departure questionnaire for these purposes at a later stage.

In the upcoming months, a series of tests will be carried out in various stages as follows.

- First of all, the questionnaire needs to be thoroughly tested and debugged before any other tests, or the validity of these may be compromised by errors in the programming. This test will be performed by people other than the programmer and ensure that all possible response options and question branches are explored.
- After the debugging phase is the Usability testing phase, the aim of which is to assess whether the questionnaire is easy to use. A usable product is one that helps users achieve their goals quickly, successfully, and with satisfaction. Jakob Nielsen, widely considered an authority on usability testing, demonstrated that 85% of usability problems could be identified from testing on the first 3 to 5 subjects, whereas 15 subjects were required to identify 100% of usability problems [8]. But instead of testing once with 15 subjects, he recommended testing on 3 to 5 subjects, modifying the application, then retesting 3 to 5 new subjects in iterative cycles until no new major problems were identified.

Typically, the test subjects will be asked to conduct an IPS survey using the tablet questionnaire. We plan to collect behavioural measures via different means such as capturing paradata but also with post-test evaluation forms.

- Paradata: for this first series of test, we only wish to capture basic information (e.g. click locations, information entered, time on tasks), therefore setting the Audit Trail to FIELD level will be sufficient. We will use Blaise API and C# to extract the paradata from various tablets used for testing. We plan to purchase a small number of different tablets (8-inches and 10-inches screen size) for our pilot. Analyses will be carried out with any suitable software package e.g. SAS, SPSS, Matlab or R.
- Post-test evaluation forms/self-report: this is used to capture qualitative information such as users' perception of the survey instrument. Criteria include: validity, reliability, internal consistency, replication and ease of use. These can also be quantitatively measured using a 5-point Likert scale: (strongly disagree, disagree, neutral, agree, and strongly agree). In terms of user experience, a good survey instrument is one that users can fill out in an efficient manner without making errors, and without getting frustrated. Therefore, accuracy is an important criteria to capture. This can be quantitatively measured e.g. the percentage of forms completed correctly, percentage of drop out and associated reason for drop out, number of errors etc.
- At the moment, there is no plan to test the questionnaire itself because we very much keep the same wording and survey flow as in the paper form, and the same interviewer scripts.

We are also currently exploring suitable ways to evaluate/measure the benefits of removing the CADI stage. In fact, we only assume that eliminating manual data keying would cut down on human errors and thus improve data quality. However, research has not yet provided evidence to substantiate the claim that computerised interviews are less error prone. If the electronic questionnaire is badly designed and difficult to use, and/or if the program has not been thoroughly debugged etc. we may even introduce new type of errors that we need to capture and adjust for. This is why the upcoming Usability tests are extremely important.

7. What next?

At the moment, there exist a number of on-going investigations the results of which will shape the design of the third prototype of the IPS questionnaire. These include:

- A health and safety check carried out by IPS field managers in Titchfield to decide on logistic aspects in particular suitable screen size of the tablet, colours to be used that will have a direct impact on the questionnaire design.
- A review of the hard/soft edit checks by the IPS research team.
- A possible streamline of the current IPS questionnaire in respect of question wording and routing. This is also motivated by the fact that some questions – as they are currently layout in the paper form – do not lend themselves very well to be routed/displayed in similar manner on a computer tablet screen. We need to investigate whether modifying their order and appearance would introduce any bias to the survey.
- Investigation of the end-to-end process to understand how the new electronic questionnaire will fit into the existing system as a replacement of the current (paper forms + CADI). As the existing system is in Blaise 4.8, some major adaptations are to be expected. In the worst case scenario, the current Blaise questionnaire may need to be reverted back to Blaise 4.8.
- A field test of the second prototype is planned for the second week of May 2015, the feedback from which will also be taken into account for the third prototype.

8. References

- [1]. Tim Burrell. The use of mobile devices to carry out Blaise surveys at the Office for National Statistics. IBUC 2007 11th International Blaise Users Conference, 2007
- [2]. User Guide (Volume 1) Background and Methodology. International Passenger Survey Overseas Travel and Tourism <http://www.ons.gov.uk/ons/guide-method/method-quality/specific/travel-and-transport-methodology/international-passenger-survey/index.html>, 2014
- [3]. User Guide (Volume 2) Background and Methodology. International Passenger Survey Overseas Travel and Tourism <http://www.ons.gov.uk/ons/guide-method/method-quality/specific/travel-and-transport-methodology/international-passenger-survey/index.html>, 2014
- [4]. Norman D. The design of everyday things. Basic Books, New York, 2000
- [5]. Biggs et al. Guidebook for Conducting Airport User Surveys. Transportation Research Board of the National Academies Airport Cooperative Research Program (ACRP) Report 26 http://onlinepubs.trb.org/onlinepubs/acrp/acrp_rpt_026.pdf, 2009
- [6]. Inc Unison Consulting. FINAL REPORT Los Angeles International Airport 2011 Passenger Survey Results and Findings. Presented to: Los Angeles World Airports, 2011
- [7]. Kathy DeBoer et al. Tablet and Web-Based Regional Airport Travel Survey with Synchronized Bluetooth Traffic Data Collection. Transportation Research Board (TRB) Annual Meeting, 2013
- [8]. Nielsen J. Why you only need to test 5 users. <http://www.useit.com/alertbox/20000319.html>, 2010.

Studies in Blaise 5

Leif Bochis Madsen, Statistics Denmark

Abstract

Statistics Denmark has carried out a couple of studies into the use of Blaise 5 since the last IBUC. The studies comprise a combination of Blaise 5 instruments and CATI Survey Management using our well-established Cati Survey and case management system and a study into using field interviewing on Android-based tablets for our Household Budget Survey. The paper describes the challenges and summarizes our conclusions from the studies.

Introduction

The properties of Blaise 5 questionnaires (separation of layout from logic, the possibility to automate layout, the support of a multitude of collection platforms, etc.) all sound very promising to Statistics Denmark – an institution that has carried out data collection using Blaise for almost 25 years.

The investments, however, in an infrastructure capable of automating the tasks of managing surveys and questionnaires and carrying data to and from the data collection system are considerable. Two studies have therefore been initiated in order to establish a number of preconditions needed to move our interview tasks towards Blaise 5.

Among the questions and preconditions we want to handle are:

- 1) To which extent is it possible to use our current infrastructure?
- 2) How large parts need to be rewritten?
- 3) How well does Blaise 5 web questionnaires work?
- 4) How well does Blaise 5 questionnaires work for CATI interviewers?
- 5) Can we replace Windows based laptops with e.g. Android based tablets for CAPI interviewing?
- 6) How easy is it to change the development from Blaise 4 to Blaise 5?

CATI Study with Blaise 5

The first of our two studies examine the incorporation of Blaise 5 questionnaire instruments in our current CATI Survey Management System. Blaise 5 – obviously – is still an incomplete product lacking support of CATI and therefore we want to examine the possible use of Blaise 4.8 CATI management together with Blaise 5 instruments. An early proof-of-concept experiment briefly after the IBUC/2013 showed that it was possible to combine a Blaise 4.8 CATI instrument with a Blaise 5 instrument, so the purpose of the study is to dig deeper into the practical details of this coexistence. Also: The vast majority of our studies then were dual mode CAPI and CAWI, so we also wanted to establish the facts about how much effort we need to invest in replacing elements of our combined Survey Management System.

CAPI study using Android based tablets

The second study emerged from another basis. The Danish Household Expenditure Survey is managed by Statistics Denmark using an external provider to carry out the interviewing. The external provider has undergone a technological move replacing pc-based CAPI interviewing and they are now carrying out their CAPI interviewing using Android based tablets. Statistics Denmark is responsible for the development of the questionnaire software, but the external provider has put pressure on us in

order to move towards the tablet based solution claiming that our survey is the only one left that is still pc-based. However, we also saw this as an opportunity to study these aspects of Blaise 5 closer.

Organisational changes

In the meantime there has been a fusion between the Interview Service Division of Statistics Denmark and the Interview Services of the Danish National Institute of Social Research bringing all activities – including a CAPI interviewer corps of a nation-wide 250 interviewers – together in a new large division in Statistics Denmark – “DST Survey”. The organisational change was announced late August 2014 taking effect from January 1st, 2015.

These changes have also emphasized the need for evaluations of the software used now and in the future in order to make strategically sound decisions for the new organisation.

The CATI Study

First step was a conversion of a Blaise 4.8 dual mode questionnaire (CATI and CAWI) for the Omnibus Survey into Blaise 5. This part comprised learning the new layout system of Blaise 5 and it definitely requires a large effort to get a thorough understanding of the different components and their interaction. Also, the layout system has evolved throughout the period we have worked with this project as new builds have been released. Therefore, while learning, we have occasionally had the feeling of aiming at a moving target. However, when finally getting a good grasp of it the layout system shows very promising with respect to automation of the creation of quite sophisticated questionnaire layouts.

Second step was making a test sample and publish as CAWI questionnaire. Because the XML files of Blaise 4.8 and Blaise 5 are compatible it was quite easy to prepare for a study. The ETL program (Extract, Transform and Load) included in the recent versions of Blaise 5 also shows promising though it would be a nice feature if more parameters were added in order to fully automate load and extract tasks (but maybe we'll have to wait for Manipula?). The Web questionnaire also works fine, though with a slower performance than Blaise 4.8.

Third step was a simulation of the CATI situation. The questionnaire works, but the testers lacked the short cuts that are available in a Blaise 4.8 DEP instrument for quickly moving through the questionnaire. This is quite an important issue and needs addressing in future releases. Of course, short cuts make less or no sense in a wide range of clients (tablets, smart phones, web browsers), but in running CATI instruments they are quite crucial for the performance of the interviewers.

Fourth step was intended to comprise inclusion of the Blaise 5 instrument into our well-established Blaise 4.8 based CATI Survey Management System. As long as Blaise 5 does not comprise CATI handling it is necessary to incorporate Blaise 5 instruments for our dual mode surveys into another CATI management system. An early proof-of-concept showed it possible to combine a general Blaise 4.8 CATI data model with an external call to a Blaise 5 questionnaire. Because of the lack of short cuts in Blaise 5 and the slightly slower performance of Blaise 5 instruments we have not carried out this part of the study.

We are considering a continuation of the study by running web questionnaires in Blaise 5 while running the CATI version in Blaise 4.8. In the long run it is not convenient to make two versions of the questionnaire but because the recent releases comprise a tool for automatic conversion of source code from Blaise 5 into Blaise 4 – and because CATI instruments need less caretaking of layout issues – it might be a possible solution for a shorter period of time.

The CAPI Study

The study was initiated in May 2014 as a suggestion from our external provider who in the previous year went through a transfer of running all CAPI questionnaires on Windows based laptops into running all on Android based tablets. At the time an Android DEP app had already been announced by Statistics Netherlands, so we assumed that it would be a reasonable schedule to aim for a pilot study in the fourth quarter of 2014 and to go into production from the new year of data collection in the middle of January 2015. If we began trying out the app, there would probably be time for the Blaise team to correct the minor bugs, we could find during the last half of the year.

Unfortunately, the Blaise DEP app for Android was withdrawn about the same time and during the BCLUB meeting in June it was announced that a release could be expected during the summer.

During autumn and winter 2014-2015 there has been several releases of the Blaise 5 package and the Android client – each one taking care of one or more crucial problems with the previous releases. Apparently, with release 5.0.2 it seems now possible to carry out a full study.

Conversion of a large Blaise 4.8 questionnaire into its Blaise 5 equivalent, however, was done quite fast. Only a few days were needed in order to make the needed adjustments to prepare the questionnaire for a Blaise 5 version to start testing. Also, this study was carried out after the other study and we could take advantage of our gained knowledge of the layout system and thus e.g. make use of automatically applied layouts.

A missing feature in Blaise 5 is the lack of support of remarks (though handling of remarks is mentioned in the documentation). Though rarely used for our CATI studies, it is quite crucial for the Household Expenditure Survey that the interviewers may add some comments to the values. The sub-optimal solution for our study has been to change the general “Amount” field type into a combined “Amount and Comment” block type. It works, but we would have preferred a general solution.

While writing these lines we have just initiated a larger scale field study and are eagerly awaiting the outcomes.

Conclusions

Both of the studies are still in their initial phases and therefore our conclusions are quite preliminary.

The development environment has brought numerous improvements, but works much slower than the Blaise 4.8 Control Centre. Also, there is definitely a steep learning curve for learning to use the layout tools.

While testing and developing – especially for the CAPI study – numerous changes have been made to the tools. Some of these changes resulted in incompatibility between the tools. Especially, proper use of the Android client has caused the necessity of simultaneous upgrade of the Blaise 5 installations on server, development computer and client tablet as well as a re-preparation of the instrument in order to make the components work together. However, it is our impression that Blaise 5 steadily matures and maybe soon we will be able to move our CAPI studies towards use of Blaise 5 and tablets.

The initial questions may only be partly answered by now, but we may summarize:

- 1) It is possible to use large parts of our current infrastructure in an initial phase. Reconsideration, however, will be needed when Blaise 5 in the future comprises implementations of CATI and Manipula.
- 2) Without rewriting we should be able to start using Blaise 5 for Web studies – also in an initial phase.
- 3) Blaise 5 web questionnaires work quite well, though performing slightly slower than the Blaise 4.8 counterparts.
- 4) Blaise 5 questionnaires lack short cuts needed for efficiently conducting CATI interviews.
- 5) We may probably be able to replace Windows based laptops with Android based tablets for CAPI interviewing, but we still await practical outcome of our study.
- 6) There is a steep learning curve regarding the layout specification when changing the development from Blaise 4 to Blaise 5. However, the benefits are very promising.

Acknowledgements

Thanks to my colleagues Saliha Zayoum and Lars Peter Jørgensen for setting up the Omnibus questionnaire in Blaise 5 and patiently working their way through all the new concepts.

Thanks to the Blaise Team for always answering quickly to our panicky questions and for moving Blaise 5 steadily forward.

Converting the Administration of Health Measures and Bio Specimen Collection from Paper to CAI

Rebecca Gatward, Karl Dinkelmann, Ian Ogden, Piotr Dworak, & Heidi Guyer,

Survey Research Center, University of Michigan

1. Introduction

The Health and Retirement Study (HRS) is a longitudinal panel study that surveys a representative sample of more than 26,000 Americans over the age of 50 every two years. Supported by the National Institute on Aging (NIA) and the Social Security Administration, the HRS explores the changes in labor force participation and the health transitions that individuals undergo toward the end of their work lives and in the years that follow.

In preparation for the 2014 wave of the HRS we successfully converted administration of the series of physical measures, bio sample collection and consent forms from paper to computer assisted (using Blaise). This paper focuses on the conversion process, focusing specifically on the following; usability issues, testing, maintaining continuity of data, acceptance by interviewers and respondents, outcomes and future development.

2. Background

The HRS interview is comprised of an in-depth CAI interview (face to face or telephone), a series of physical measurements, collection of biomarker samples and at the end of the interview a self-administered paper questionnaire is left with the respondent which they are asked to complete and return by mail.

Investigations around the collection of Physical Measures and Biomarker samples on the HRS began with feasibility tests in 1992 and were introduced fully in 2006. Collection of this type of data conducted by interviewers is now becoming more common across social surveys. For the past few waves the type of measures collected on the HRS has remained stable and is comprised of the following:

- Physiological or performance assessments
 - Blood pressure
 - Lung strength
 - Grip strength
 - Balance
 - Walking speed
- Anthropometric measurements
 - Height, Weight, Waist circumference
- Biomarkers samples
 - Saliva
 - Blood (dried blood spots)

Administration of these tests occurs around two thirds of the way through the interview and, until the most recent wave of the study (2014), was administered using a paper booklet. The design of the booklet was improved each wave as we learned more about the type of interviewer instructions we needed to include and as technology advanced – for example, the latest versions used by interviewers included

scannable barcodes which were added to reduce the risk of data entry errors and subsequent linkage issues.

The Physical Measures and Biomarker booklet consisted of 34 pages and included three consent forms (separate consent is collected for the physical measures, blood and saliva sample collection). Eligibility for, or adaptations of the tests are based on age and responses provided in the interview or quality of samples provided in the previous wave. At the beginning of the booklet the physical measurements and Bio (PM and Bio) markers are listed and interviewers transferred information about eligibility from a summary page in the Blaise questionnaire to the paper.

Interviewers then started the section by gaining signed consent for the series of physical measures and proceeded to administer the first set of measures. Each measure in the booklet followed a common format – including a reminder of the equipment required to administer the measure, an introductory paragraph the interviewer read to the Respondent which described the measure, some safety questions and then the step by step instructions. After each measure, interviewers would record any measurement data and then complete observation questions. Following the approved protocol is essential and including the step by step instructions is one of the methods used to ensure consistency between interviewers administration of the measures. Finally, interviewers completed the data entry as a post interview task once they returned home – transferring the data from the paper booklet into the Blaise questionnaire and then mailing the completed PM and Bio booklets, including the signed consent forms, to Ann Arbor to be logged and scanned.

The design of the booklets has developed over the waves as technology advanced and we gained more experience through the quality assurance processes. For example, the version used during the 2012 data collection wave included preprinted serial numbers and barcodes which were scanned into the Blaise questionnaire.

3. Converting from paper to CAI – design goals and objectives

The following section describes the goals we followed during the process of developing and designing the PM and Bio section.

3.1 Minimize Respondent burden

The primary goal of the conversion was that we should not increase the interview length by administering the PM and Bio section using CAI and entering the data at the time of the interview. We needed the design of the section to enable interviewers to administer the section as efficiently as possible and simplify the process where we were able.

3.2 Maintain data continuity

HRS is a longitudinal study so it was essential to maintain data continuity. We could not make any changes which could affect the time series data for respondents or the data structure. The variable names had to remain the same and any additional variables had to be kept to a minimum.

3.3 Single mode for in-person data collection

One of the major aims of the conversion was to move all data collection into one mode. Administering the PM and Bio markers in a single mode would save time and money across the survey process. Savings would be realized through a reduction in printing costs, paper handling, post data entry, eliminating the time required by interviewers to prepare mailing the booklets, no postage costs, no booklets to log on arrival at the central office, reduction in storage space required for paper materials and a reduction in the number of cases requiring reconciliation between paper booklet and Respondent.

3.4 Electronic consent administration

Converting the PM and Bio section also provided the opportunity to transition to electronic consent. This would provide the same cost savings around paper handling and reconciliation resulting from moving from the paper booklets. Electronic consent forms would also further improve adherence to protocol and ensure that signed consent was obtained prior to beginning the measurements. One requirement in this transition was not to reduce consent to participate in the measures. There were some concerns that respondents would be concerned about the electronic signatures.

Additionally, a new application of capturing the electronic consent from respondents was created. This application is a standalone C# program that is called via Alien procedure. The procedure then uses Manipula to call the consent applications executable and import information needed (such as the name of the consent file to use, the name to save the sign consent as, etc). Figure 1 below shows an example of the consent form that is administered on the laptop.

Figure 1: Electronic Informed Consent Form

The screenshot shows a web-based electronic consent form titled "HEALTH AND RETIREMENT STUDY PHYSICAL MEASURES CONSENT FORM". The form is displayed within a browser window titled "Electronic Documentation Utility (EDU) V5.7.2.0". The form content includes:


- Study ID: H4A00061128, IRB: Health Sciences and Behavioral Sciences, Date Approved: 1/04/2014.
- Introduction: You have already agreed to participate in the Health and Retirement Study (HRS) funded by the National Institute on Aging and conducted by the University of Michigan. The study director is Dr. David Weir of the Survey Research Center. The study will help provide a very complete picture of how people 50 years of age and older living in the United States are faring as they head into retirement and enjoy their retirement years.
- Physical Measures: In addition to completing the questionnaire, you are now being asked to complete some physical measurements. The physical measures conducted will allow researchers to better understand the connections between health status and other indicators of interest such as economic and employment status.
- Agreement: If you agree, you may be asked to complete up to 8 different physical measurements which involve standing, walking, exhaling, gripping an object with your hands and having your blood pressure, height, weight, and waist measurements taken. It will take approximately thirty (30) minutes to complete these physical measures.
- Results: You will be sent a copy of your blood pressure results. These results are not a substitute for a doctor's evaluation. If your results are above the normal range, you will be encouraged to share this information with your doctor.
- Benefits: Although you may not receive direct benefit from your participation, others may ultimately benefit from the knowledge obtained in this study.
- Risks: There are no known risks associated with completing this portion of the study. In the unlikely event of any injury resulting from the research, no reimbursement, compensation, or free medical treatment is offered by any of the co-sponsors of this research project.
- Incentive: There is no additional incentive for completing these measures, nor is there a penalty if you choose not to complete this component.
- Confidentiality: As with all Health and Retirement Study data, the results of the physical measurements will be de-identified before being made available to researchers. You will not be identified in any reports on this study. There are some reasons why people other than the researchers may need to see information you provided as part of the study. This includes organizations responsible for making sure the research is done safely and properly, including the University of Michigan, government offices or the study sponsors.
- Reporting: We are required to report to state officials credible evidence of serious harm or abuse to any person, but these measures do not involve such topics.
- Confidentiality Certificate: A Department of Health and Human Services Certificate of Confidentiality covers this research in order to help ensure your privacy. This certificate can help protect the investigators from being forced to release any research information that identifies you.
- Voluntary Participation: Participating in this component of the research study is completely voluntary. Even if you decide to participate now, you may change your mind and stop at any time.
- Questions: If you have questions about your rights as a research participant, or wish to obtain information, ask questions or discuss any concerns about this study with someone other than the researcher(s), please contact the University of Michigan Health Sciences and Behavioral Sciences Institutional Review Board, 540 E Liberty St., Ste 202, Ann Arbor, MI 48104-2210, (734) 936-0933 [or toll free, (866) 936-0933], irb@hsb@umich.edu. Please refer to protocol HUM00061128.
- Agreement: By signing this document, you are agreeing to participate in the physical measures component of the study. You will be given a copy of this document for your records, and one copy will be kept with the study records. Be sure that questions you have about the study have been answered and that you understand what you are being asked to do. You may contact us if you think of a question later.

Below the text, there is a section for signatures and dates:


I agree to participate this component of the study.

Please type Respondent's name:

Please sign here and date

 Date:

Interviewer's Signature and date

 Date:

At the bottom, there are "Save" and "Cancel" buttons.

Once the consent form is saved in the application, the program saves the signed consent form as a PDF file and writes information about the form to a text file. A secondary Alien procedure is called by Blaise to read this consent information back into the Blaise instrument. This allows the Blaise instrument to know if it is okay to continue administering the current physical measure or bio marker collection.

3.5 Minimize reliance on interviewer decision making

Using paper booklets meant that interviewers had to make decisions or handle routing throughout the PM and Bio administration. Obviously, in CAI the routing, range checks, consistency checks, signals and hard checks could all be handled by the Blaise questionnaire and interviewers would be prompted to take action as necessary. Alleviating the burden and reliance on interviewers is particularly important in the PM and Bio section as they have numerous tasks to perform and it is essential that study protocol are followed. One example of a study protocol which relied on an interviewer taking appropriate action is identifying Respondents with high blood pressure. Interviewers are required to provide a card to the Respondent which recommends that they see a physician if their blood pressure is above a certain level. Further action is necessary if their Blood Pressure is at the level of a possible Hypertensive Crisis, in this situation the interviewer must alert the Respondent and ask if they feel they are able to proceed with the interview. CAI provided the opportunity to provide the alert to the interviewer which reduces the risk of an interviewer not following these critical protocol.

Administering the PM and Bio section using CAI also eliminates the need for interviewers to decide if the Respondent is eligible for certain tests. Again, this allows them to concentrate on following the correct protocol and reduces the risk of administering inappropriate measures. One particularly complicated decision point is the hand strength test where there are four potential paths through the measure depending on the respondents dominant hand or if they have any issue which restricts their use of one hand.

3.6 Data entry at the interview location

Moving data entry from a post interview process to happen within a Respondents home provides an opportunity to resolve any error checks which may be triggered in consultation with the Respondent.

4. Solution and challenges

In the next section of the paper we will describe the design of the CAI PM and Bio section and the challenges we faced during the conversion process.

4.1 User interface

We decided to keep the format of each measure very similar to the structure of the paper booklet. Each section started with a reminder to interviewers about what equipment was needed to administer the section, followed by an introduction or description of the measure to be read aloud by the interviewer, a demonstration, safety/eligibility questions, a setup checklist, the measurement and then post measurement observations. This provided a consistent structure for each measure and some familiarity for experienced interviewers who had used the paper booklets. Using a consistent structure, we hoped, would also contribute to the process being as efficient as possible.

In the CAI version we were also able to provide an on screen cue to indicate where the interview is in the administration process.

A simple header to indicate progress was added to the top of each screen:

Intro – Demo – Safety – Setup – Measure-- Obs

Bolded text indicated where the interview is in the process:

Intro – Demo – **Safety (① of ②)** – Setup – Measure-- Obs

Figures 2 and 3 below provide examples of how this appeared on screen. Throughout the section, we presented headers indicating the current measurement which was also a progress indicator for the interviewer.

Figure 2: Example of Header in the Blood Pressure Measurement

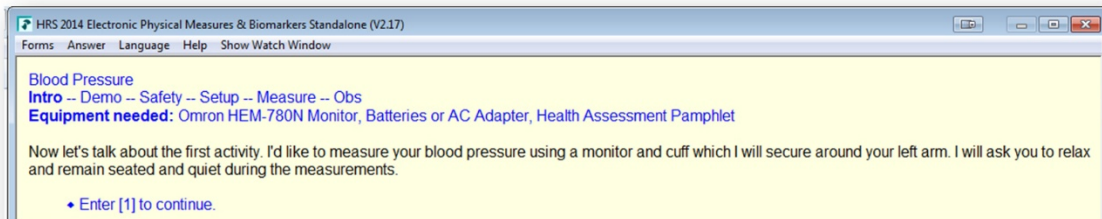


Figure 3: Example of Header and Instructions in the Saliva Sample Collection



4.2 Single mode for data entry and administration

One challenge we faced was to meet the goal of designing a PM and Bio section which could be administered in a single mode. The physical logistics of administering this section was a challenge, particularly for the measures where the interviewer has to move away from where they are seated for the interview to conduct specific measures – such as the height, weight and walking speed measures. For these measures we realized that it was impractical for the interviewer to take their laptop with them as they administer them or for the interviewer to return to their laptop to enter data between measures. We decided that the most efficient option was to provide a booklet for the interviewer to administer the height, weight, waist and walking speed. The data from these measures is then entered at the end of the PM and Bio section before proceeding with the interview. We do not require interviewers to return this booklet to the central office, as they do not record any identifiable information on the form they are allowed to recycle the booklet. Data entry for this portion takes a short amount of time, and interviewers are instructed to pass the Respondent a pamphlet which includes more detail about how the data from these measures are used and they can review the blood pressure results, which the interviewer records in the pamphlet. This also provides a good opportunity for Respondents to take a short break before continuing the interview.

4.3 Use of media

Converting the section to CAI also allowed us to include media – such as videos and photos within the questionnaire both for Respondent and interviewers reference. Photo reminders were placed within the section to remind interviewers of the correct placement of equipment (figure 4). We were able to use videos and photos to demonstrate the measures to Respondents, rather than the interviewer

doing so. Using videos also helped standardize the demos and explanatory introductions provided by interviewers (figure 5). We also used timers – which were videos – to time the correct waiting time between some measures which were repeated multiple times (figure 6). These eliminated the need for interviewers to use a stop watch. The timings were triggered automatically as the data entry for the previous measure was entered, which, again, reduces interviewer burden. The ‘stop on key’ functionality was disabled which forced the entire video to run before proceeding to the next question.

Figure 4: An Image Used to Aid in the Hand Strength Measurement



Figure 5: Videos to Demonstrate the Walking Speed Measurement

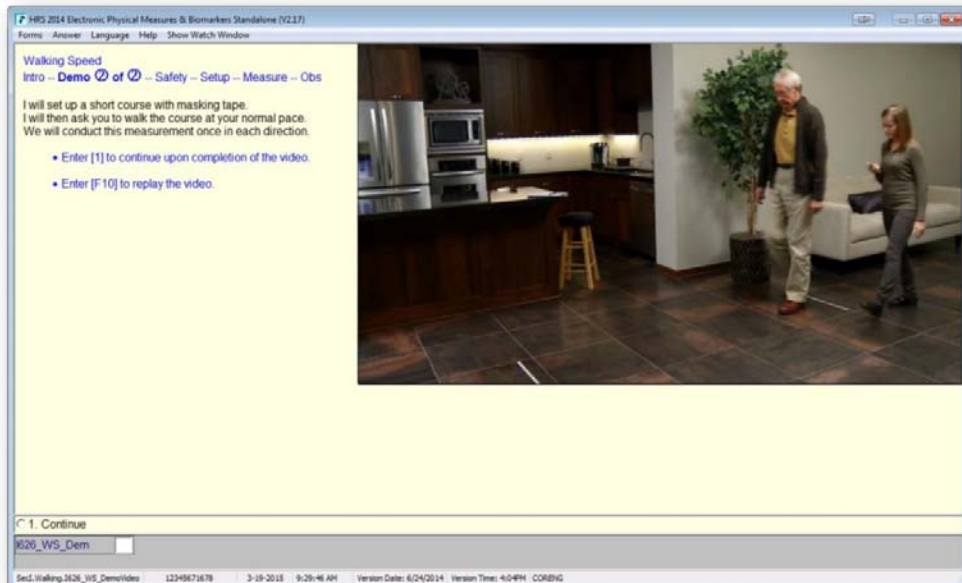
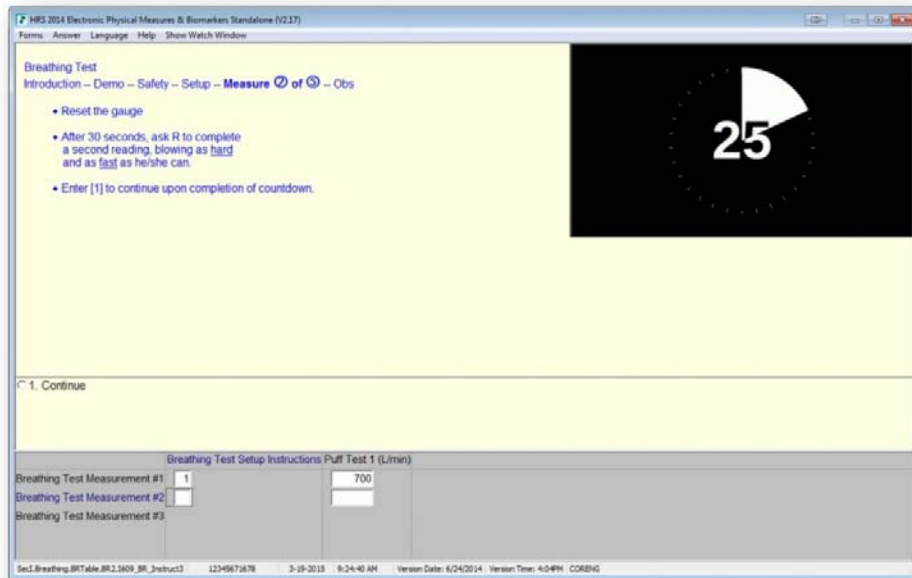


Figure 6: Video Timers to Ensure Proper Wait Time between Measurements



4.4 Reduction in fields requiring data entry

We were also able to take advantage of the capability in CAI to auto fill fields. These included various completion flags and timings.

4.5 In production revisions

Although avoided, CAI does allow the flexibility to make in production revisions. Making similar changes using paper booklets would be costly, lead to version control issues and would only be made if absolutely necessary.

4.6 Text formatting and screen layout

The step by step instructions provided to interviewers for some measures are extensive, ranging from around eight to 20 steps (an example is provided in figure 7 below). For those measures with more steps it was necessary to split the instructions across multiple screens, they could not be split arbitrarily so some time had to be spent deciding how to group the steps.

Figure 7: Blood Sample Instructions

HRS 2014 Electronic Physical Measures & Biomarkers Standalone (V2.17)
Forms Answer Language Help Show Watch Window

Blood Sample
Intro -- Demo -- Safety -- **Setup 11 of 11** -- Measure -- Obs

- 1) Put on a pair of vinyl gloves.
- 2) Clean the R's finger with the alcohol prep pad and make sure the R's finger is dry before proceeding.
- 3) Remove blue tip from lancet in order to activate it. While holding the R's hand firmly, place the lancet on the side of the pad of the R's middle or ring finger or the R's thumb.
- 4) Prick the finger with the lancet. If necessary, instruct the R to gently squeeze his/her finger from the base several times to form a large drop of blood.
- 5) With the sterile gauze pad, wipe away the first drop of blood.
- 6) When the next large drop of blood is formed, allow the blood to drop onto the center of the left-most circle on the first blood collection card.
- 7) Once the R completely fills one of the circles, assist the R with filling the other two circles on the collection card and then fill the three spots on the second card for storage.
- 8) If the R is able and willing, fill the remaining two spots on the first card and then continue to fill the remaining two spots on the second card.
- 9) Provide the R with a bandage.
- 10) Place the used gauze pad, alcohol wipe and wrappers in the plastic bag.
- 11) Place used lancet into the small biohazard container.
- 12) Place both blood spot cards into the cardboard box with flaps tucked behind and spots facing outward. Allow a minimum of two hours of drying time.
- 13) Remove gloves and put in same plastic bag, then take home to discard.
- 14) Keep bag of shipment materials, loaded drying box and completed lab-authorization form together. After cards have dried, you will prepare the samples for shipment.
- 15) Refer to the Blood Sample Packing and Shipment Instructions sheet for shipment instructions.

• ENTER [1] to continue.

1. Continue

1657_Bld_Instru

Sec1.BloodSpot.1657_Bld_Instru12 12345671678 3-19-2015 9:35:42 AM Version Date: 6/24/2014 Version Time: 4:04PM CORENG

4.7 Allow out of sequence completion

From experience administering the PM and Bio section, we knew that it was sometimes necessary to administer some measures, particularly the Bio samples out of sequence so it was necessary to allow for this in the CAI version. For example, some Respondents are unable to provide the saliva sample in the time allowed. In this situation, the interviewers could delay entering details of the saliva sample until later in the section. This was accomplished by programming the saliva questions within a procedure where the questions were presented as auxiliary fields and then saved back to their original location when Blaise comes out of the procedure. Similar flexibility had to be provided to handle a situation where the interviewer was unable to use the bar code scanner to input a serial number. In this situation we had to allow manual entry (although this was discouraged, unless absolutely necessary). Figure 8 and 9 below, provides screenshots of the relevant screens).

Figure 8: Option to Manually Enter Barcodes

HRS 2014 Electronic Physical Measures & Biomarkers Standalone (V2.17)

Forms Answer Language Help Show Watch Window

Saliva Sample

Intro -- Demo -- Safety -- Setup -- Measure 3 of 4 -- Obs

Equipment needed: Barcode Scanner

Using barcode scanner, scan ID from the Saliva barcode label in this field. Press [Enter] to continue to the saliva collection tips, which may be shared with the respondent.

- ENTER [993] if barcode scanner is unavailable.
- ENTER [999] if a saliva sample will not be collected.

Enter a text of at most 13 characters

933_SalvaB 993

Sect.Saliva.1933_SalvaBarCodeID 123456789 3-19-2015 9:33:20 AM Version Date: 6/24/2014 Version Time: 4:04PM CORENG

Figure 9: Three Follow Screens to Capture Manual Barcode Entry

HRS 2014 Electronic Physical Measures & Biomarkers Standalone (V2.17)

Forms Answer Language Help Show Watch Window

Saliva Sample

Intro -- Demo -- Safety -- Setup -- Measure 4 of 4 -- Obs

Barcode ID Manual Entry: ###-AAAA-###

- Record first three digits of Saliva ID without dashes or spaces

Enter a numeric value between -99 and 999

941_SalBar

942_SalBar

943_SalBar

Sect.Saliva.1941_SalBarCodeIDMan1 123456789 3-19-2015 9:33:13 AM Version Date: 6/24/2014 Version Time: 4:04PM CORENG

4.8 Interviewer acceptance

One of the major considerations throughout the conversion process was to design a tool which interviewers liked to use. This meant that the design had to be good – the version we launched had to be well thought out and with minimal ‘loose ends’. We involved interviewers by collecting feedback throughout the conversion process and referred back to feedback provided on previous occasions when the idea of converting to CAI was considered. The conversion process was an iterative testing process which started prior to the 2012 data collection when we carried out a small pilot of a first CAI version of the section. This test involved a small number of interviewers and the section included

some of the functionality, such as video timers and photos which were included in the final version, but with some refinements based on interviewer feedback.

5. Outcome

Overall, the launch of the new CAI version of the PM and Bio section has been successful. Many of the goals have been met and we have gained efficiencies through administering the section using CAI – these are discussed in further detail in this section of the paper.

5.1 Interviewer feedback

Prior to the interviewer training at the beginning of the 2014 wave, we had anticipated that the experienced interviewers may be skeptical about moving from the paper booklet to CAI. Around 40 per cent of approximately 200 HRS interviewers involved in the 2014 data collection had experience using the paper booklet. We were pleased, however, that experienced interviewers accepted the move to CAI and during training, some interviewers expressed a preference to administration using CAI rather than the paper booklets.

We found that training the experienced interviewers was more about training them how to adapt to the CAI administration and feel comfortable with the tool. For example, we needed to remind interviewers not to jump ahead of the steps displayed on each screen but to move on to the next screen once they had completed the last step on the screen.

Even after training and part way through the data collection, a few interviewers still felt that administering the section using a laptop created a barrier between them and the Respondent in what was a section they could have a break from the laptop and be more engaged with the Respondent.

Overall interviewers became more comfortable during the data collection wave – data collected from interviewers showed that they were more positive about the section after the first few months of data collection. The following are a selection of feedback from interviewers:

Pros...

Also, feel that Respondents are more engaged in electronic administration. Writing results in booklet felt somewhat like a judgment or diagnosis.

[in the Blood Pressure section] Rs liked seeing the timer; Rs asked to see computer screen and wanted to see the demonstration images. Rs interested in seeing the screen. More screen-sharing.

Four of my Respondents commented on transition. "Must be saving the trees"

Felt like it went more quickly;

I love the opportunity to lessen the chance for error. No chance for transcription error. Also, psychologically less draining as there are fewer post-interview tasks to complete.

The flow just seems to work smoothly on the computer. Easier to just enter the data into the computer once instead of recording it on paper then having to transfer it to the computer later.

Cons...

The Interviewer is locked into the screen and not engaged with the Respondent and doesn't let the Interviewer be untethered from the laptop.

5.2 Consent improvements

During the 2014 wave the consent rates for the physical measures, blood and saliva samples have been consistently higher than previous waves. We also made some changes to the training for 2014, which may also have influenced this increase, but we are confident that at least some of this improvement is due to the conversion of the consent forms from paper to electronic administration. Interviewers have provided feedback that the electronic form is less intimidating to Respondents than a paper form.

5.3 Interview length

The primary goal of the conversion was not to increase interview length. The timings data suggests that administering the section using CAI has not lengthened the time spent on this section with the respondent and has reduced the overall time required to administer and enter the data for this section.

5.4 Streamline reconciliation

The time spent on reconciliation has been greatly reduced both for the paper booklets and the consent forms.

6. Further development

We will use feedback from interviewers to further improve the design of the PM and Bio section as we prepare for the next wave of data collection which will begin in 2016. Along with specific issues mentioned by interviewers, we aim to eliminate the need for an administration booklet for the four measures which remain on paper.

We may also consider reviewing and adding some touchscreen aspects similar to those used by Dinkelmann and colleagues (2015) to aid the administration of some of particular areas of the PM and Bio section. Additionally, employing this touchscreen approach could potentially allow the interviewers to do the entire section on the computer. We may also explore using native camera as the barcode scanner or adding checks to ensure they are using their barcode scanner.

7. References

Dinkelmann, K., Guyer, H., & Gatward, R. (2015). *Prototyping Touchscreen Blaise Applications*. The Sixteenth International Blaise Users Conference (IBUC), Beijing, China.

Programming Blaise for a Multi Questionnaire Study

*Youhong Liu, Heidi Guyer
Survey Research Center
the University of Michigan*

1. Introduction

The Panel Study of Income Dynamics (PSID) is a longitudinal survey of a representative sample of U.S. men, women, children and the families in which they reside. Data on employment, income, wealth, housing, food expenditures, transfer income, and marital and fertility behavior have been collected annually or bi-annually since 1968. From 5,000 families in 1968, the study has grown to include over 10,000 families. In all interactions with our study members, we use the name the Family Economics Study (FES).

The Child Development Supplement (CDS) was initially fielded in 1997; the U.S. had very limited nationally representative longitudinal studies of children focusing on child health and development in the preadolescent years. CDS was designed to increase our knowledge about how these and other changes are influencing child development and well-being.

The original CDS followed a cohort of children in PSID families who were 0–12 years of age in 1997 through three waves of data collection and focused on understanding the socio-demographic, psychological, and economic aspects of childhood in an on-going nationally representative longitudinal study of families. In 2014, all of the children in the original cohort have reached adulthood, and a new generation of children has replaced them in PSID families. The goal is to collect information in 2014 on all children aged 0–17 years in this new generation, shifting the orientation from a cohort study to one that obtains information on the childhood experiences of all children in PSID families. The sample will consist of approximately 6,400 children aged 0-17 and 3,500 primary caregivers.

In this paper, we will discuss the Blaise implementation for the most current wave of CDS, conducted in 2014.

2. CDS 2014 Respondents and Blaise Instruments

There are three kinds of respondents: COVERSCREEN, THE CHILD and PRIMARY CAREGIVER (PCG).

With these three kinds of respondents, the Study is defined with three Blaise Instruments. In the chapters below, we will discuss the Blaise programming techniques used for each of these instruments.

3. Coverscreen Instrument

The Coverscreen is used to verify information about the preloaded CDS children and other family member to identify who each child's primary caregiver (PCG) is. The program then generates the appropriate number of sample lines for each respondent.

The Coverscreen is the instrument preloaded for each household in the sample. Since it is used to obtain information that determines the respondents as well as the structure and the flow of all the other portions of the interview, it must be completed before any of the PCG or Child sample lines are spawned. Once the Coverscreen is completed, it cannot be revised or edited without intervention from the Ann Arbor office. This is why it is extremely important that it is programmed accurately. If data are entered incorrectly in the Coverscreen, the consequences can be costly:

- a. Interviewing Family Units (FU) where the CDS child is actually ineligible
- b. Not interviewing a FU where the CDS child is eligible
- c. Interviewing the wrong PCG
- d. Passing bad data to subsequent interviews
- e. Incorrect age and grade of the CDS child impacts many skip patterns in all of the interviews.

In order to collect the information in the Coverscreen effectively and robustly, project staff designed several challenging features for Blaise programmers to implement. A few examples are provided in the following sections.

3.1 Children's relationships to others in the household

The Coverscreen instrument starts with the household members table which verifies members names, birthdays and FU status. The result can be complicated in households with many children and/or family members, as shown below:

Figure 1: Family Member List

Family Listing

Review the family listing with Respondent. If necessary, go back to make changes. Once past this screen, you will not be able to go back to the Family Listing

Respondent: Doug Davis

Other Family Members:

- Donna Davis, Age 50, Moved out to live on (his/her) own
- Kevin Davis, Age 1, Moved out to live with someone else (CDS Child-Eligible)
- Jake Davis, Age 3, Moved out to live with someone else (CDS Child-Eligible)
- Austin Davis, Age 6, Living in the household; (CDS Child-Not Eligible)
- Kara Davis, Age 7, Living in the household; (CDS Child-Not Eligible)
- Sarah Davis, Age 10, Living in the household; (CDS Child-Not Eligible)
- David Davis, Age 11, Living in the household; (CDS Child-Not Eligible)
- Brad Davis, Age 14, Living in the household; (CDS Child-Eligible)
- Alex Davis, Age 15, In the military (CDS Child-Eligible)
- Anna Dans, Age 16, Away at school (CDS Child-Eligible)
- Brian Dans, Age 18, Living in the household; (CDS Child-Not Eligible)

♦ ENTER [1] to confirm family listing and continue

1. Continue

CDS Respondent

CS Confirm

Following the household member table is the relationship table. The purpose of this table is to identify the CDS children's relationship to other family members, aged 16 or older, who can

potentially be the PCG respondent of a CDS child. The table is loaded with all CDS children as columns and potential PCGs in the family as rows. If a CDS child is 16 or older, then he/she is also be selected a PCG. In order for users to navigate the Blaise DEP easily and complete the table quickly, here are some of the requirements from the research staff:

- In the same row, if a later child's relationship to a FU member is the same as a previous child, then the relationship question will be skipped and a relationship is auto assigned.
- In different rows, if a child is a sister, brother or cousin of a previous child, then the relationship question is skipped, and relationship is auto converted and assigned.
- Children list in some of the questions is generated dynamically.

To meet the above requirements, complex programming techniques are implemented. As many as four For/Do loops are used in the section. Keeps on fields and blocks are used so data can be passed properly from parent loops to children loops.

Figure 2: Relationship Table – Code 17 (Sister) is Detected and Converted to Code 13 (Brother)

Child	Doug	Austin	Kara	Sarah	David
Doug	5	1	4		
Austin	5	5	95		
Kara	5	6	13		
Sarah	5	7	13		
David	5	8	17		

Figure 3: Code Example – Convert Relationship between Sisters, Brothers and Cousins

```
xRICPrev := EMPTY {Start empty}
FOR I:=1 TO 24 Do
  IF I<pFmIndex THEN { Loop Previous Rows}
    FOR J:=1 TO Num_CDSKid DO {Loop all columns}
      IF RICFmLstB[I].RITCCDSLstB[J].RITCCSAQSN = pRICFMAQSN {AQSNs are FU member's IDs}
        AND RICFmLstB[I].RITCFMAQSN = RITCCSAQSN THEN
          IF RICFmLstB[I].RITCCDSLstB[J].RTC >=BSister AND RICFmLstB[I].RITCCDSLstB[J].RTC<=Cousin THEN
            {Biological/Step/Half/Adoptive Sister/Brother or Cousin }
            xRICPrev := RICFmLstB[I].RITCCDSLstB[J].RTC
            xGender1 := RICFmLstB[I].RITCCDSLstB[J].RITCCSGender
            xGender2 := RICFmLstB[I].RITCFMGender
          ENDIF
        ENDIF
      ENDDO
    ENDIF
  ENDDO
  IF xRICPrev<>EMPTY THEN {Found a match}
    {Procedure Assigning code to the current cell based on previous Gender and Relationship}
    AssignSibRelCode (xRICPrev, xGender1, xGender2, RTC)
    RTC.SHOW
  ENDIF
ENDFOR
```

3.2 Preload Spawning

As previously described, accurate completion of the Coverscreen is very important. It is a driver of the two other Blaise instruments: PCG and Child. These two instruments have very complex preload data structures themselves. Data is pulled from different parts of the Coverscreen and used for fills and logic in the PCG and Child instruments.

There are over 100 preload variables for the Child instrument and 600 preload variables for the PCG instrument. Additionally, the number of Child and PCG lines generated from a Coverscreen can vary by household. In fact, Coverscreen lines can also be derived from an original Coverscreen, if a child is moved out with or without an adult.

a. Preload Generation

For other studies at the University of Michigan, the sample management system is responsible for pulling out and generating the preload information for spawned lines. Given the complex preload data structures for CDS 2014, it is difficult for a process outside of Blaise to figure out which data goes to which instrument, and what is the logic if a code conversion is needed. It was determined that the best way to generate the preload was to generate the strings inside the datamodel with Blaise code only. The Coverscreen's programmer can perform string creation relatively easily and accurately because of their knowledge of the variable locations and logic.

The actual generation of the preload strings is easy to understand in Blaise. It is done by concatenating variable values, e.g. PreloadStr:=Field1 + Field2. The challenging aspect of generating the preload string is to match and interpret variables between the Coverscreen and the spawned datamodels. With careful design, programming, several rounds of modifications and testing, we found this process was very effective.

Figure 4: Code Example – Child Datamodel Preload Generation

```
Locals
  SCount : INTEGER {for special kid}
RULES
  xTEMP := PCGKids[pPCGIndex].CDSKids [pKidIndex].KidIndexInFU
  xTEMP1 := PCGKids[pPCGIndex].PCGIndex
  xTEMP2 := PCGKids[pPCGIndex].CDSKids [pKidIndex].KidIndex

  PCGPre [pPCGIndex].KidPre[pKidIndex].KidFName := CS3Table.HHLListBLOCK[xTEMP].CS3FName
  PCGPre [pPCGIndex].KidPre[pKidIndex].KidLName := CS3Table.HHLListBLOCK[xTEMP].CS3LName
  PCGPre [pPCGIndex].KidPre[pKidIndex].KidAQSN := CS3Table.HHLListBLOCK[xTEMP].CS3AQSN
  PCGPre [pPCGIndex].KidPre[pKidIndex].KidPSIDAQSN := CS3Table.HHLListBLOCK[xTEMP].CS3PSIDAQSN
  PCGPre [pPCGIndex].KidPre[pKidIndex].KidCDSAQSN := CS3Table.HHLListBLOCK[xTEMP].CS3CDSAQSN
  PCGPre [pPCGIndex].KidPre[pKidIndex].KidSampleID := SampleID + 'P' + str(pPCGIndex) + 'C' + Str(pKidIndex)
  xTempPre := ''

  xTempPre :=xTempPre + PCGPre [pPCGIndex].KidPre[pKidIndex].KidSampleID + '^' {1}
  xTempPre :=xTempPre + pXTempS {2-11, passed in from PCG}
  xTempPre :=xTempPre + str(Elig.ChildEligB[xTemp].ChildElig.ORD) + '^' {12,Preload.ChildElig}
  xTempPre :=xTempPre + CS3Table.HHLListBLOCK[xTEMP].CS3PSIDAQSN + '^' {13,Preload.CHPSIDAQSN}
  xTempPre :=xTempPre + CS3Table.HHLListBLOCK[xTEMP].CS3AQSN + '^' {14,Preload.CHAQSN}
  xTempPre :=xTempPre + CS3Table.HHLListBLOCK[xTEMP].CS3CDSAQSN + '^' {15,Preload.CHCDSAQSN}
  xTempPre :=xTempPre + Preload.PRELOADHHLTable.PRELOAD_HHL [xTEMP].Six8ID + '^' {16,Preload.CH68ID}
  xTempPre :=xTempPre + Preload.PRELOADHHLTable.PRELOAD_HHL [xTEMP].PN + '^' {17,Preload.CHFN}
  xTempPre :=xTempPre + str(Preload.PRELOADHHLTable.PRELOAD_HHL [xTEMP].RTH.ORD) + '^' {18,Preload.CHRTH}
  xTempPre :=xTempPre + str(CS3Table.HHLListBLOCK[xTEMP].CS3FU.ORD) + '^' {19,Preload.CHFUHU}
  xTempPre :=xTempPre + CS3Table.HHLListBLOCK[xTEMP].CS3FName + '^' {20,Preload.CHFName}
  xTempPre :=xTempPre + CS3Table.HHLListBLOCK[xTEMP].CS3LName + '^' {21,Preload.CHLName}
  xTempPre :=xTempPre + str(CS3Table.HHLListBLOCK[xTEMP].CS3Index.ORD) + '^' {22,Preload.CHIndex}
```

b. Preload Spawning Testing

With the above preload spawning method we were in need of a way to test spawned lines. Our sample management system can perform some integration test, but it is difficult to test different scenarios and combinations within it. During the project development, it became apparent that we needed a feature within our CAI Testing Tool (CTT) that would allow us to spawn lines. This new feature was required for multiple reasons, including:

- Ensure that the number of lines generated is correct
- Ensure the number of preload variables for each datamodel generated is correct
- Confirm values are pulled from the correct Coverscreen fields and are loaded into correct spawn line fields
- Use the generated preload strings to test the Child and PCG instruments, to avoid manual data entry

After reviewing the existing CAI Testing system, we determined that this feature could easily be added.

- In the Coverscreen datamodel, a variable named SpawnData is added, and its question text is a well formatted xml string that can be parsed by the testing system. The xml string consists of preload string fields in the datamodel
- In the testing system, a C# module is developed to exam the xml string and to pull out the value of the fields specified from the Coverscreen and then to generate lines for PCG and Child datamodels This C# module is mainly developed for the CDS study but it can also be used for other studies if the datamodel is programmed accordingly

Figure 5: XML String for Spawn Line

```
AUXFIELDS
SpawnData "<SpawnLine>
  <SpawnDetail cttStudyID = 'CDS14' cttpid = ""SRC.SRO.CDS14.CS"" desc=""Split off"">
    <FieldName loop = ""6"">CSBLOCK.PreOutSO.SOPre[x].Str_</FieldName>
  </SpawnDetail>
  <SpawnDetail cttStudyID = 'CDS14' cttpid = ""SRC.SRO.CDS14.Child"" desc=""Child"">
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[1].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[2].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[3].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[4].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[5].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[6].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[7].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[8].KidPre[x].Str_</FieldName>
    <FieldName loop = ""15"">CSBLOCK.PreOutPCG.PCGPre[9].KidPre[x].Str_</FieldName>
  </SpawnDetail>
  <SpawnDetail cttStudyID = 'CDS14PCG' cttpid = ""SRC.SRO.CDS14.PCG"" desc=""PCG"">
    <FieldName loop = ""9"">CSBLOCK.PreOutPCG.PCGPre[x].Str_</FieldName>
  </SpawnDetail>
  <SpawnDetail cttStudyID = 'CDS14' cttpid = ""SRC.SRO.CDS14.Child"" desc=""Not Elig"">
    <FieldName loop = ""15"">CSBLOCK.PreOutNotElig.NotEligPre[x].Str_</FieldName>
  </SpawnDetail>
</SpawnLine>": 1..1 {Xml string for CAI testing system to spawn lines}
```

c. Sample Management System Preload to Spawned Blaise Datamodel

With careful preload generation and testing, the preload task for our sample management system is simplified greatly. A small number of variables are pulled, the values can be saved

in text files and then loaded into the spawned lines by a Manipula script as with the other projects.

4. Parallel Block Programming

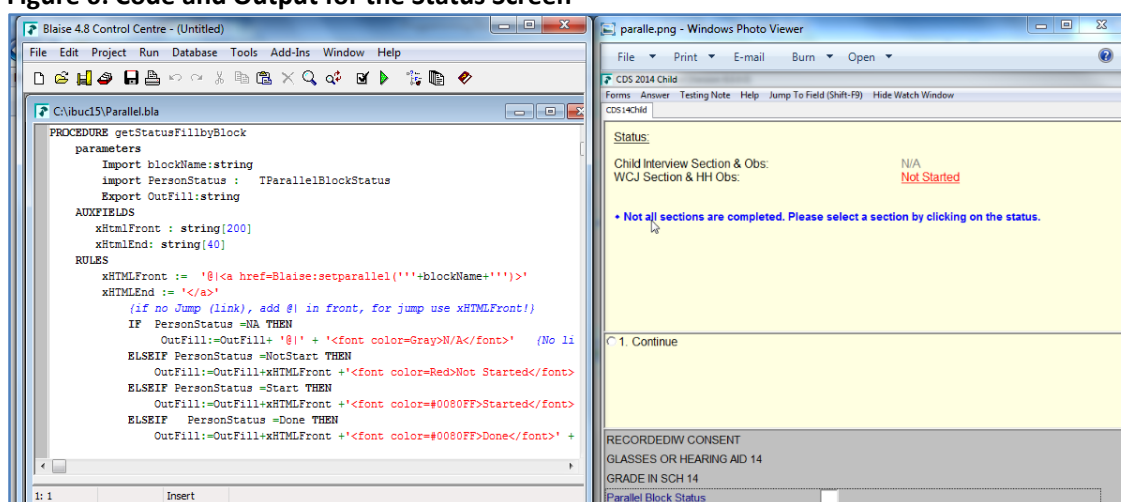
Parallel Block programming enables you to process one or more block fields separately from the current route in the DEP. This technique is widely used in the Blaise community. CDS 2014's Child and PCG datamodels have employed this technique in two ways:

4.1 Access Parallel Blocks on a Status Screen

Based on our previous experience, we were aware that we needed to be extremely cautious about parallel programming method. If it is not programmed properly, wrong statues may be assigned and result in not the main block or some of the parallel blocks not completing as required. To ensure proper programming, we implemented the following:

- All parallel statue generations are made in one procedure instead of in the main body of the datamodel
- Not every block has the same set of statues. We examined and documented the details in the procedure carefully to ensure proper assignments are made
- On the statues screen, startparallel procedure is used to jump into parallel blocks
Extreme caution was taken during the programming since some statues can jump, while others cannot jump
- Color codes are used for different statues so it is easy for interviewers to distinguish

Figure 6: Code and Output for the Status Screen



4.2 Parallel Blocks to Enable and Disable the Achievement Tests

There are three achievement tests in the Child and PCG datamodels. It is important that we allow a respondent to stop at any point of a test. To accomplish this, another parallel block feature was implemented inside of a main parallel block (WCJ). There are three sub-parallel blocks used to start and stop the three tests respectively.

These blocks are made visible on the Blaise parallel tabs by adjusting the BXI file. Again, they should be programmed carefully so the tabs are shown for the right tests and data are saved properly for interrupted tests.

Figure 7: Code Example – a Parallel Block for Enable and Disable a Block

```
LWStart
LWask.KEEP (Grade) {Letter word test questions, keep is used to save both complete/incomplete test data}
LWEND.KEEP
StopBLOCK.KEEP {Parallel}
IF LWStart<>EMPTY AND LWEND=EMPTY THEN {Prevent Iver to go back to change data if the test is finished LWEND<>EMPTY}
  IF StopBlock.StopLW <> Stp then {Stop test if the stop button is checked in the parallel block}
    LWask (Grade) {In the testing block}
  ENDIF
ENDIF
LWEND
```

5. Woodcock Johnson Assessments Programming

The Woodcock Johnson (WCJ) Test of Achievement is a well-known, established educational assessment tool. CDS has three subtests. They are:

- Letter-Word Identification
- Passage Comprehension
- Applied Problems

In the test, the term “basal” is used to refer to when the respondent obtains six or more consecutive items correctly. The term “ceiling” is used when the respondent answers six or more items incorrectly. Blaise programming is required to automatically calculate the basal and ceiling using the codes entered by interviewers and determine what items should be given at what time.

In previous waves of CDS, all of these tests were programmed and worked properly. For the recent wave, we decide to modify the programming so that it is modularized and can be easily understood by another programmer if a problem occurs and needs to be fixed. Time was spent to ensure that one test was implemented correctly, and we then applied the method to the other two tests. The core technique is to eliminate the hard code and use one procedure for basal and one procedure for ceiling calculations. To be able to call the procedures in different tests and different items, parameters are passed into the procedure. In the end, the basal and ceiling procedures were called as many as eighty times. Additionally, both the PCG and Child instruments were able to share the same code with very little modification. This method was much appreciated in terms of maintenance and troubleshooting by the project team

6. Conclusion

CDS 2014 is a complex study. Several challenging features were specified by the research staff in order to assist interviewers in easily navigating the various data collection instruments while collecting quality data. With careful design, testing and implementation, new features were developed to program three high quality instruments, each of which is well accepted by the research staff, project team and interviewers.

From our experience, Blaise proves to be a powerful and flexible survey program system and helped us to achieve almost all complex requirements for this study.

7. References

Blaise 4.8 Online Assistant – Statistics Netherlands

CDS 2014 Study Guide – Survey Research Center, Institute for Social Research, the University of Michigan

Alternate order of blocks for the CPS-ASEC Instrument

Roberto Picha, U.S. Census Bureau

1. Introduction

The Current Population Survey's Annual Social and Economic Supplement (CPS ASEC) generates widely used estimates on detailed income data and benefit programs. Concerns of "fatigue effect" were raised in regards to the way the information was collected. This fatigue may lead respondents to lose concentration or motivation when answering long questionnaires, affecting the reliability on the data collected. After researching and completing cognitive testing on paper, it was determined that an alternate approach should be tested in the automated instrument. The idea was to collect age and incomes related information first and then tailor the order of detailed income questions based on the information collected. Instead of asking the questions in the benefit blocks in the same order for all respondents, the sponsor wanted to alternate the order in which the blocks were asked to optimize the responses. Three alternate paths were identified: low income, seniors, and all others. This paper discusses the implementation of a research paper and its findings into an automated instrument.

The Current Population Survey (CPS), sponsored jointly by the U.S. Census Bureau and the U.S. Bureau of Labor Statistics (BLS), is the primary source of labor force statistics for the population of the United States. The CPS is the source of numerous high-profile economic statistics, including the national unemployment rate, and provides data on a wide range of issues relating to employment and earnings. The CPS also collects extensive demographic data that complement and enhance our understanding of labor market conditions in the nation overall, among many different population groups, in the states and in substate areas.

2. Background

In March 2013, the Technology Management Office (TMO) authoring team began working with the Income Source Sections and converting them in an alternated order for CPS's ASEC. The new requirement was the result of the research conducted by Westat and documented in their paper "Cognitive Testing of Potential Changes to the Annual Social and Economic Supplement of The current Population Survey." The instrument went into production during February-April 2014; and will return to production in February-April 2015.

The CPS-ASEC instrument administers up to twenty-three possible sections for participants in a household composition for ages 15 and above, and concerns of "fatigue effect" were raised in regards to the way the information was collected in the instrument. Westat's paper resulted in cognitive testing which in turn prompted the CPS sponsor to implement this alternate approach in the current automated instrument. The idea was to collect age and household level income related information first and then tailor the order of questions based on the information collected. Instead of asking the questions in the benefit section blocks in the same order for all respondents, the instrument would administer them in the order the research indicated was most appropriate for the respondent (alternate order). The order was based on information already collected and would optimize the responses.

The Table 1 below displays the order of the instrument sections based on the income type. The table shows the default, low income, and senior paths along with the order of the sections within each path.

Table 1. Alternate Order of Income Source Questions					
Default		Low Income		Seniors	
Order		Order		Order	
1	Unemployment and Workers Compensation	1	Unemployment and Workers Compensation	2	Disability
2	Disability	7	Public Assistance / TANF	3	Social Security
3	Social Security/SS for Children	8	Food Stamps (SNAP)	4	Supplemental Security Income (SSI)
4	Supplemental Security Income (SSI)	9	WIC	5	Veterans
5	Veterans	10	School Lunches	6	Survivors
6	Survivor Benefits	11	Public Housing	13	Pensions
7	Public Assistance / TANF	12	Energy Assistance	14	Annuities
8	Food Stamps (SNAP)	2	Disability	15	Retirement Accounts – Withdrawals or distributions
9	WIC- no amount collection	3	Social Security	16	Other Income Earning Assets (outside of retirement)
10	School Lunches- no amount collection	4	Supplemental Security Income (SSI)	17	Property Income
11	Public Housing- no amount collection	5	Veterans	1	Unemployment and Workers Compensation
12	Energy Assistance	6	Survivor Benefits	7	Public Assistance / TANF
13	Pensions	13	Pensions	8	Food Stamps (SNAP)
14	Annuities	14	Annuities	9	WIC
15	Retirement Accounts (within) – Withdrawals or distributions	15	Retirement Accounts – Withdrawals or distributions	10	School Lunches
16	Other Income Earning Assets (outside of retirement)	16	Other Income Earning Assets (outside of retirement)	11	Public Housing
17	Property Income	17	Property Income	12	Energy Assistance
18	Education Assistance	18	Education Assistance	18	Education Assistance
19	Child Support	19	Child Support	19	Child Support
20	Alimony	20	Alimony	20	Alimony
21	Financial Assistance from friends or relatives	21	Financial Assistance from friends or relatives	21	Financial Assistance from friends or relatives
22	Other Income I (as in ASEC now)	22	Other Income I (as in ASEC now)	22	Other Income I (as in ASEC now)
23	Other Income II (as in ASEC now)	23	Other Income II (as in ASEC now)	23	Other Income II (as in ASEC now)

3. The goals for the CPS-ASEC redesign

The initial stage of the CPS-ASEC redesign, from the standpoint of specifications, suggested following the Research Triangle Institute concept that was implemented in the Survey of Health Insurance Program Participation (SHIPP) instrument using a global counter flag to trigger specific paths. In other words, the specifications for the CPS-ASEC redesign would be defined for each section for each of the three paths using the Specialized Instrument Design Resource (SPIDER) application. This tool is used to quickly create an instrument specification in the U.S. Census Bureau.

Furthermore, from the beginning the intent was to reuse the existing code of the ASEC as much as possible. The sponsors hoped that TMO authoring would make small internal modifications in the code. While this was a great goal, the authoring team later learned that some blocks would have to be split and parameters would need to be modified. This was initially a concern that later changed once there was a better understanding of the task.

With these two goals in mind, the authors developed a small prototype to assure the sponsoring division that one set of specifications could be used to simplify the process of managing the questions despite the path taken. And, more importantly, to determine if the alternate order of sections would be feasible. There were additional changes in the sections due to new questions being added as well as the removal of other questions. Parameters no longer in use had to be removed. Data dependencies across sections were considered and modifications were expected.

3.1 A Simple approach sometimes is not that simple

The TMO authoring team looked into the routing groups containing the sections. However, there was a need to come up with a proof of concept for placing sections in specific order based on some universe. The twenty-three sections were classified into five groups (see Table 2 for the grouping suggested).

For the first attempt, the authors created a small-scale instrument; it mimicked the groups on route. Table 2 summarizes the groups containing the sections


Table 2. Grouping sections	
o	Group A Section 1
o	Group B Sections 2,3,4,5,6
o	Group C Sections 7,8,9,10,11,12
o	Group D Sections 13,14,15,16, 17
o	Group E Sections 18,19,20,21,22,23

Eventually, Section 1 was removed from Group A. The questions for this section were removed or placed into other sections. Therefore, there were only four groups that had to be re-ordered to meet the new requirements for the CPS-ASEC instrument.

Table 3 shows the three paths for the four groups. This grouping helped to simplify the final structure of the instrument, and at the same time it helped to identify the issues up front. Group E was always asked at the end regardless of the path taken. In essence, the alternate order would be between Group B, C, and D.

Table 3. Identifying the path	
○ Default path:	B, C, D, E
○ Low Income path:	C, B, D, E
○ Senior Path:	B, D, C, E

Table 4 shows a simple Datamodel that was used to prototype the concept by routing four variables representing the four groups to simulate how the sections would be used in the final approach. This simple exercise demonstrated to our sponsor that the reuse of existing blocks was not as simple as expected. The re-ordering was not a mere one step modification in the instrument that could simply be tested and placed into production.

Table 4. Simple Datamodel
<p>DATAMODEL Inst "Routing questions"</p> <p>FIELDS B, C, D, E: STRING PATH : (D "default", L "low Income", S "Seniors")</p> <p>RULES</p> <p>PATH IF PATH = D THEN B C D ELSEIF PATH = L THEN C B D ELSE B D C ENDIF</p> <p>E</p> <div>  [Error] INST2.bla(135,6): Field already displayed, or display order conflict </div>

As any savvy Blaise author would guess, the first error thrown by the compiler was "*Field already displayed, or display order conflict*". This message means that you cannot route the sections in different order.

Routing questions in Blaise can be challenging, especially if the order might vary at runtime. The restriction is that the layout cannot be changed at runtime.

Other techniques could be applied to obtain the desired result, but these techniques would not be automated. For example, parallel blocks could be used in the instrument, and the interviewers could manually select the appropriate section to interview based on the order required. The sponsors did not want interviewers determining the order in which to interview blocks. So, they proceeded to develop specifications in SPIDER to create three separate blocks, each representing one of the question order treatments.

3.2 Prototyping the alternate blocks in three paths

A second small Datamodel prototype was developed using a technique conducive to achieving the alternate order of income source questions. Expanding the Datamodel to contain the 23 sections and wrapping the sections into five defined groups (as previously described) made more sense. This also helped the analysts to visualize the future structure of the instrument.

The prototype established the concept of using a master block. The master block was the location where data was stored from any of the three paths taken in the case. This master block would aid the analyst during testing to verify that data was stored in one place and always the same place regardless of the path taken. It would also simplify the back-end data processing.

Table 5 displays the block structure for the supplement. While not fully functional for data collection, the prototype served as a model for our sponsors so they could develop the specifications required for this instrument. Essentially, the introduction of the master block concept allowed the sponsor to develop one set of specifications, and the approach was cleaner. The dependency across sections was a concern. However, that eventually subsided as the instrument for the most part was referencing variables directly as opposed to being properly parameterized.

Table 5 - Datamodel displaying the blocks

```

DATAMODEL Inst
FIELDS

  Path : (Default "B,C,D,E", LowIncome "C,B,D,E", Seniors "B,D,C,E")

BLOCK BlkB ; BLOCK BlkC ; BLOCK BlkD ; BLOCK BlkE

BLOCK BOrderOne { default order }
  FIELDS  B : BlkB;  C : BlkC;  D : BlkD;  E : BlkE
ENDBLOCK

BLOCK BOrderTwo { low Income order }
  FIELDS  C : BlkC;  B : BlkB;  D : BlkD;  E : BlkE
ENDBLOCK

BLOCK BOrderThree {Senior order}
  FIELDS  B : BlkB;  D : BlkD;  C : BlkC;  E : BlkE
ENDBLOCK

BLOCK BMaster {master copy of the answers (no matter which order)}
  Fields  B : BlkB;  C : BlkC;  D : BlkD;  E : BlkE
ENDBLOCK

AUXFIELDS
OrderOne : BOrderOne
OrderTwo : BOrderTwo
OrderThree : BOrderThree
DoOnce : 0..1

FIELDS
  Master : BMaster

```

This new structure more accurately resembled the final product. This new concept would give us the following benefits:

- **Centralized location of data** - The data collected from these various blocks would reside in a “master block”. Here the data processors could extract the data items from one location.
- **Minimize Database Size** - Using temporary blocks for data collection will not add any more space in the survey database.
- **Allows for Rostering** - The prototype was designed for one person. However, it was later adjusted to handle a household roster by adding a roster to loop through the household members.

Table 6 shows the routing and synchronization of the blocks. The first step was to copy data from the master blocks into the appropriate routing blocks. After data was collected in the routing blocks, it was copied back into the master blocks.

Table 6. Displaying routing and assignments

RULES

Master.KEEP {keep the master set of answers}

path

DoOnce.KEEP

IF DoOnce = EMPTY **THEN** {use block copy to transfer master responses to the block to be used}

IF path.ord =1 **THEN**

OrderOne.B := Master.B

OrderOne.C := Master.C

OrderOne.D := Master.D

OrderOne.E := Master.E

ELSEIF path.ord =2 **THEN**

OrderTwo.C := Master.C

OrderTwo.B := Master.B

OrderTwo.D := Master.D

OrderTwo.E := Master.E

ELSEIF path.ord = 3 **THEN**

OrderThree.B := Master.B

OrderThree.D := Master.D

OrderThree.C := Master.C

OrderThree.E := Master.E

ENDIF

DoOnce :=1

ENDIF

IF path.ord=1 **THEN** {ask the blocks in alternate order}

OrderOne

Master.B := OrderOne.B A {use block copy to transfer the responses back to master}

Master.C := OrderOne.C

Master.D := OrderOne.D

Master.E := OrderOne.E

ELSEIF path.ord=2 **THEN**

OrderTwo

Master.C := OrderTwo.C {use block copy to transfer the responses back to master}

Master.B := OrderTwo.B

Master.D := OrderTwo.D

Master.E := OrderTwo.E

ELSEIF path.ord=3 **THEN**

OrderThree

Master.B := OrderThree.B {use block copy to transfer the responses back to master}

Master.D := OrderThree.D

Master.C := OrderThree.C

Master.E := OrderThree.E

ENDIF

3.3 Final steps before implementation

Only one set of specifications in SPIDER was necessary for any path taken by the instrument. The sponsor agreed to one important business rule. Once the path was established during data collection, the interviewer was not allowed to back up and modify data that would change routing. This particular rule was stressed during training. Although, there is a solution for situations where an interviewer may change the path, it was preferred not to implement this solution as the enhancement might have added a performance issue to the instrument.

4. Implementation

Once the sponsor was satisfied with the prototype, the next step was to implement the prototype methodology into the CPS-ASEC instrument by modifying the existing code. Each of three groups containing the twenty-three sections was self-contained, but a few of them had relationships across sections. This imposed some challenges that required additional code modifications. Some of the modifications needed were:

- Changes to add and remove parameters.
- Splitting blocks
- Referencing arrayed variables directly
- The removal of fill constructs
- Layout in the modelibrary was adjusted as new questions were added

4.1 Re-use of existing code

Since the CPS-ASEC was already coded, the authors quickly understood that they needed to break some larger blocks into smaller blocks to achieve the instrument's proper routing of questions.

Examples of the changes made in the call of sub blocks are shown below.

- **Old Code:**
 - **Call to source**
bVet_Pmt({IMPORT} CTRLNUM, IN_NUMHOU, IN_RESPLI, IN_NAME, IN_SEX, IN_LAST_YEAR)
 - **Person Level:**
bVet_Pmt_Person[I]({IMPORT} IN_CTRLNUM, I, IN_RESPLI, IN_NAME[I], IN_SEX[I], IN_LAST_YEAR)
- **New code:**
 - **Call to source**
bVet_Pmt({IMPORT} CTRLNUM, IN_NUMHOU, IN_RESPLI, IN_NAME, IN_SEX, IN_LAST_YEAR)
 - **Person level:**
bVet_Pmt_Person[I]({IMPORT} IN_CTRLNUM, I, IN_RESPLI, IN_NAME[I], IN_SEX[I], IN_LAST_YEAR)
 - **Call to amount:**
bVet_Pmt_Amt({IMPORT} CTRLNUM, IN_NUMHOU, IN_RESPLI, IN_NAME, IN_SEX, IN_LAST_YEAR, IN_Q60a88, IN_Q60b_88)

As you can see, there were really no differences between the old method and the call to the source blocks in the new method.

The only changes occurred in the call to the Amounts blocks when using the new path. Additionally, inside the person level blocks in the new method there were many instances where new parameters were generated (GPs). This is because the fields that needed to be accessed were person level source fields and passing them would mean passing (in the case of veterans questions) four parameters per person and in some blocks more. In the old method, this was not necessary since the fields were at the same level.

4.2 Unexpected Issue

As the work on the redesign took shape, one unexpected major issue made us change the approach. Data was not being correctly saved to the Auxfield blocks as expected.

During our system test, the subject matter team observed that when exiting the case in the middle of the new income sections, data was lost. While data was stored properly in the master block, it was noted that upon re-entry of a previously touched case, the data did not appear to copy into the Auxfields blocks. This occurred particularly in the Amount blocks where the logic relied on data from the source block to be on route. However, because of the way Blaise works (Select Check Mechanism), it was not properly evaluating the parameters referenced causing the amount data to be thrown off path.

Because there were several sections and blocks involved, the amount of work it would take to correct the issues by redesigning the communication between sections was significant. An executive decision was made to make a more conservative change in the instrument. Instead of using Auxfields for the alternate order of blocks, these Fields would be converted into regular Fields.

This change brought some consequences for the post collection processing stage.

4.3 Comparison of Meta file

Since the Auxfields blocks were converted to Fields blocks, the size of the instrument also increased. Now the CPS ASEC instrument became the largest Blaise instrument compiled in the U.S. Census Bureau. To have an idea in the increment, the size of the metafile went from 2,856KB to 18,764 KB.

4.3.1 Technical description of the model instrument

The following tables (7 and 8) show the differences of the metafile size before and after the redesign. The increase in the size is significant. Using Cameleon and the technical description of the metafile for the CPS-ASEC before and after the redesign shows the increment of the metafile size.

Table 7: Overall counts	Values		Percent
	ASEC 2013	ASEC 2014	Increase
Number of uniquely defined fields*1	2,473	5,681	229.7%
Number of elementary fields*2	2,329	5,182	222.5%
Number of defined data fields*3	24,623	69,131	280.8%
Number of defined block fields*4	144	499	346.5%
Number of defined blocks	233	368	157.9%
Number of embedded blocks	9	9	100.0%
Number of block instances	1,570	4,580	291.7%
Number of key fields	1	1	100.0%
Number of defined answer categories	928	1,321	142.3%
Total length of string fields	191,205	396,662	207.5%
Total length of open fields	0	0	na
Total length of field texts	143,863	424,282	294.9%
Total length of value texts	43,343	50,153	115.7%
Number of stored signals and checks	10,069	25,428	252.5%
Total number of signals and checks	10,069	25,428	252.5%

Table 8: Data fields	ASEC 2013		ASEC 2014	
	Number	Length	Number	Length
fields in data file				
Integer	11167	39582	28466	116226
Real	323	2213	323	2213
Enumerated	7346	8811	23216	25769
Set	507	790	3523	6198
Classification	0	0	0	0
Datatype	53	424	53	424
Timetype	3	24	3	24
String	5224	191205	13547	396662
Open	0	-	0	-
Total in data model	24623	243049		

5. Struggles in Post-Collection Processing CPS-ASEC

The Current Population Survey Programming Branch (CPSPB) was an active participant in the process of the new redesign. For the purpose of data output, the sponsoring division defined the specifications for these sections only once and not three times. In the initial design of the instrument, with the use of temporary Auxfields, data processing appeared to be straightforward. However, due to the later change that required Auxfields to be converted to Fields, data processing became much more complex. The CPSPB soon discovered that there were duplicates of all their records. The duplicate records appeared in the raw data. Upon reviewing the data, it was noted that the duplicates were instances of the blocks and the instance was not always the same one. Because this was unexpected, CPSPB was puzzled and unsure when extracting the data from case to case. The reason behind this duplicate data traced back to the blocks that were reused for data collection and storage. Therefore, their instance would be reflected in output. The first, second, and third instance would represent the data of the path used for the case. The fourth (last) instance was in fact the master block and its data.

The CPSPB followed specifications from SPIDER that was used to process the CPS-ASEC instrument. Below is a description of the method used. The main task was identifying the blocks that had these duplicates in them. These references, (i.e., instances of the blocks) were found during an internal 50 case test. This test is a step used internally to validate the data before the instrument is deployed to production. The method applied to validate the test is also the same method used for production data processing.

The following is a high level description of what the CPSPB were looking at. These are the files from an ASCII Relational dump.

- CPS-ASEC test output consists of many text files (labeled inst.A01, inst.A02, etc), called blocks.
- Each block is either a person level block (consists of person records) or a household level block (consists of household level data).
- Each block has a different file layout, which is documented in our .RAS file. However, the first 13 columns for EVERY block are identical:
 - FPRIMARY 1 – 8
 - INSTANCENUMBER 9 - 13
- Fprimary is the record ID and Instance Number represents the income path taken.
- For Household Level blocks:
 - 1 = default income path
 - 2 = low-income path
 - 3 = senior income path
 - 4 = output record (can represent any path)
- For Person Level Blocks:
 - 1-16 = default income path
 - 17-32 = low-income path
 - 33-48 = senior income path
 - 49+ = output record (can represent any path)
- There was a misunderstanding of records with an Instance Number = 1, 2, or 3. The same information from one of these records was duplicated in Instance Number 4 (household level blocks). The data processors were not sure where to retrieve the data from. In this case, the record is duplicated (two identical records in a block exist, and only their Instance Number is different) so the data processors were instructed to always use the Instance Number 4 for retrieving the household level block data.
- The same output was noted for Person blocks, except that now the Instance Numbers will be 1-16, 17-32, 33-48, and 49+. The data processors were instructed to retrieve their data from Instance Numbers 49-64 for the person level output records.

5.1 What CPSPB would want instead?

The CPSPB that processes the CPS data desired a slight change in the Blaise output to produce simpler, more straightforward data in their ASCII Relational dump - specifically the data of the instance number of the path taken for the case.

For household blocks, the data output team had to remove all records that had an Instance Number = 1, 2, or 3. These are all duplicates of the Instance Number=4 records.

For person blocks, the data output team had to remove all records that had Instance Number < 49. These are all duplicates of the Instance Number 49-64.

For example:

Household level block B03

<u>Fprimary</u>	<u>InstanceNumber</u>	
00000001	1	← remove
00000001	4	
00000002	2	← remove
00000002	4	
00000003	2	← remove
00000003	4	

Person level block B34

<u>Fprimary</u>	<u>InstanceNumber</u>	
00000001	1	← remove
00000001	49	
00000002	17	← remove
00000002	50	
00000003	29	← remove
00000003	49	

To meet the requests of the data output team, we could reinstate block Auxfields with some other modifications. Then only one Instance would be present in the data for the household level and one for each person level, regardless of the order of collection.

5.2 How Data is handled in Output

Original raw data are not actually removed from records; the CPSPB removes the records as they process it. The sponsoring division tells the CPSPB which blocks these duplications are occurring in and provides instructions on which records to remove via specifications.

This step tells the CPSPB which blocks have duplicate records. However, for household blocks, the CPSPB has to go a step further. They must use the .RAS file to find out where the inst.xxx blocks are in SPIDER. For example, see below.

B33: blkUWComp_Source
 B35: blkInc_Source_A
 B50: blkInc_Source_B
 B54: blkInc_Source_C
 B64: blkInc_Source_D
 B71: blkUWComp_Amt
 B73: blkInc_Amt_A
 B74: blkDis_Incom_Amt

Since the sponsor uses SPIDER to produce a list of variables it wants on output via physical location, this location matches the RAS file produced. As explained earlier, the extra instances threw off the post data collection processing.

5.2 Master Control System (MCS) struggles in the process of recycling cases

Another issue was found during production. Our Master Control System (MCS) experienced issues when running a Manipula recycle script against the CPS ASEC metafile. The script was used to recycle cases from CATI to CAPI and vice versa. When the MCS ran the script, an access violation crashed their process. The instrument and scripts were compiled using Blaise 4.8.1 build 1403. Per Statistics Netherlands advice, MCS modified their process to use Blaise 4.8.4 build 1861 Manipula to process the Blaise 1403 Manipula script. This change resolved the access violation. While we were not able to identify the actual issue that caused the crash, TMO and the sponsoring division decided to move the instrument to a newer version of Blaise for 2015 production.

6. Next Steps

The CPS-ASEC is currently in another round of interviewing for 2015. There were no major changes made to the ASEC other than the typical changes such as text, fills, and adding/removing questions. The structure established from the Redesign implemented in 2014 will be similar for 2105. However, the CPSPB has modified their process to read the proper instance blocks in order to simplify their processing.

- The 4th instance for house hold data
- The 49th + instance for person level data

6.1 Going back to Auxifields

Late in 2014, TMO Authoring attempted to revert to Auxifields blocks with few adjustments to the original approach. The initial findings indicated that some parts of the income blocks may lose data. This happened to all blocks that were arrayed. Adding some Keep statements might have provided a solution. However, there was not sufficient time to test this change adequately. We plan to research a solution for this issue for the 2016 data collection effort.

6.2 Manipula can be handy in the future

For 2016, we plan to synchronize the store and load using Manipula. The proposal is to insert the script in the instrument. We still need to add some keeps to the blocks that initially were not retaining the data. The goal is to reduce the size of the metafile and make the instrument more robust while still maintaining the quality of the data collected.

7. Conclusions

Although we were unable to implement the desired design for the CPS ASEC supplement requirements, we were still able to implement the desired requirements by our sponsors. Overall, the mission was accomplished, but not in the desired manner. We, of course, learned from this experience.

- It was much more challenging than expected to implement a new instrument design into existing instrument code. It was very tedious to break up existing income questions blocks into smaller blocks so they could flow as required.
- Larger instrument tend to generate new issues in production.
- Returning to the Auxfields approach by using Manipula and adding proper keeps to the code should help to reduce the size of the instrument and simplify the data processing by eliminating duplicate data.
- Output data should be reviewed and analyzed by the data processors in the development stage and not after production has started.
- Because the sponsoring division fielded the CPS ASEC instrument in 2014 by itself, there was no way to measure how the redesigned ASEC supplement affected the quality of the data. As a result, management decided to field a second instrument in March 2015 in parallel with the (redesigned) production data collection effort. The second instrument consisted of the 2015 basic CPS instrument with the old (i.e., pre-redesign) ASEC supplement module. The sponsoring division will compare the data from the two instruments and draw conclusions as to the effectiveness of the redesigned supplement.

9. Acknowledgment

I'd like to acknowledge the following authors and analyst for their input into this paper.

- Johanna Rupp, ITS specialist. TMO
- Lisa A Cheok, survey statistician. ADDP
- Tim Marshall, survey statistician. ADDP

10. References

Wendy Hicks and Jeffrey Kerwin, "Cognitive Testing of Potential Changes to the Annual Social and Economic Supplement of The current Population Survey," July 25, 2011.

Barbara S. Bibb, Lillie Barber and Ansu Koshy, "Coding Tricks to Save Resources, Research Triangle Institute (RTI)," USA IBUC 2010 13th International Blaise Users Conference

The views expressed in this paper are those of the authors and not necessarily those of the U.S. Census Bureau.

Consumer Expenditures Proof of Concept: A Joining of Quarterly and Diary

Michael K. Mangiapane – U.S. Census Bureau

1. Introduction

The Consumer Expenditures Survey project (CE) is a long-standing survey conducted by the U.S. Census Bureau under the direction of the U.S. Bureau of Labor Statistics (BLS). It originally started out in the late 1800s as a survey that was conducted approximately every ten years until 1980. The data obtained from the survey is used to update an important economic indicator known as the Consumer Price Index (CPI) which measures price inflation in the U.S. economy. Due to a need for more frequent data about consumer spending patterns and faster updates of the CPI, BLS changed CE to be a continuously running survey made up of two instruments. These instruments became the Consumer Expenditures Quarterly (CEQ) and the Consumer Expenditures Diary (CED). While these instruments have been serving the data collection needs of BLS, they are always looking for new and improved ways to collect expenditure data. In this paper, we will discuss what could become the next generation of CE instruments and the steps we are taking to achieve BLS's needs.

2. Design and History of CE Instruments

CEQ and CED interview separate samples of households in the United States. A household that falls into either sample is interviewed as a single group of people, also known as a Consumer Unit (CU). If there is anyone living at the household that pays their own living expenses, they are interviewed as a separate CU and added to the sample. A household that falls into the CEQ sample is interviewed every three months for a total of five interviews (this was revised in February 2015 to four interviews). Each interview consists of collecting data about the expenditures that the CU had made for the past three months. This instrument is designed to capture data about large expenditures that a respondent would recall easily, such as the cost of rent or a mortgage, spending on furniture purchases, vacations, car payments, and educational expenses. A household that is a part of CED is visited two or three times during a three-week period. A brief interview is conducted with the CU before they are given a paper diary to record all of their expenditures for one week. The first diary is picked up and a second one is given to the CU to record another week of expenditures, for a total of two weeks of recorded expenditure data. The idea behind CED is to gather data on expenditures that a CU may not recall in a regular interview (e.g., CEQ) such as a visit to a convenience store or a single purchase at a fast-food restaurant. The only expenditures that are not included in either survey are business-related expenditures or expenditures where the CU was reimbursed.

Both CEQ and CED started out as PAPI surveys. Since there are many different types of expenditures that a CU can make, CEQ became a very large survey. Before its conversion to CAPI, a CEQ form would have a minimum of 76 double-sided ledger-sized (11 inch by 17 inch) double-sided pages to be filled out. If there was more than one person in the CU or they owned more than three properties, extra pages would have to be added to the appropriate sections. CED's form was made up of 10 double-sided ledger-sized pages of questions about the household along with a section to record any diary entries that may have been missed by the CU but recalled when the Field Representative (FR) returned to collect each diary. The diaries themselves were 26 double-sided letter-sized pages. As one can imagine, with the

costs of printing and processing each of these instruments, the CE surveys were excellent candidates to be converted to CAPI.

In 2000, an initiative was undertaken by the Census Bureau and BLS to convert CEQ and CED from PAPI instruments to CAPI instruments using Blaise to implement these instruments. The use of Blaise was significant because CEQ and CED were two of the first instruments to be developed in Blaise at the Census Bureau. After a period of development and testing, CEQ went into production in the field in April 2003 and CED went into production in January 2004.

2.1 Current CE Instruments

Currently CEQ is running in Blaise 4.8.4.1861 and it contains twenty-two different expenditure sections. Additionally, there are sections in the instrument to collect the demographics of the CU and the characteristics of the household. After the demographics are collected, the FR asks questions about the CU's expenditures for the past three months, including related follow-up questions like if purchased items were bought for the CU or someone outside the CU and if taxes were included on the reported expenditures. Sections that ask about income are asked only during the second through fourth interviews, and a section about financial assets is only asked on the fourth interview. There is also a debriefing section in the back of the instrument that collects information about any records that were used during the interview and any aids that may have been used, such as an information booklet or a home file. All available sections in the instrument for the interview are accessible on the main path or they can be accessed via parallel tabs, giving the interviewer the ability to jump to specific sections during the interview. CEQ has kept a similar design for CAPI as it had for the PAPI version, but with the flexibility to collect more expenditure data during an interview. However, the CEQ instrument does make for long interviews with respondents. The average interview requires 1 – 1.5 hours to complete. This may have an effect on response rates for the survey. Additionally, moving to CAPI has allowed the sample size to increase. Since 2005, approximately 5,000 households in the U.S. are interviewed every month for CEQ.

CED is running in Blaise 4.8.4.1861 and contains a nearly identical demographics section as CEQ. Once that information is collected, the CU is asked a small subset of questions such as if they own their home, what they spend on groceries, and how many cars they own. There is also a section that asks about each CU member's employment status and income information, just like in CEQ. That section is optional for the first two visits, but it must be completed during the final visit. The FR leaves a paper diary for the CU to record their expenditures for a 1-week period. After the initial week, the FR picks up the first diary and leaves a second diary for the CU to fill out, and the FR asks some follow-up questions about the first diary to prompt the CU to recall any expenditures they may have potentially forgotten to record. For the final visit the FR will pick up the second diary and collect income information from the CU if that has not been done already. Approximately 2,000 households are interviewed for CED every month.

While CED is a smaller instrument than CEQ, it does have some unique requirements that required being creative when we coded the instrument. The instrument is date-sensitive in that the FRs are given a window in which they are to make each of their visits. The first window is provided from an input file and the FR has six days to make initial contact with the household and leave a diary for the CU. After the diary is placed, the instrument calculates a date one week in the future and gives the FR and the respondent a seven-day window to pick up the first diary and leave the second one. After the second visit, the instrument calculates another date one week in the future and creates a seven-day window for

the FR to make their final visit. There are also times where it may not be feasible for an FR to make three visits to the CU because of travel distance, workload, or because the CU knows they will be unavailable during one of the visit windows. The FR is allowed to do what is called a “double-placement” in that they place both diaries with the CU on the first visit. In this scenario the instrument calculates a date that is fourteen days in the future and creates a seven-day window for the FR to return. The instrument also has to skip what is considered to be the “normal” second visit and combine all follow-up questions to be asked on the final visit. All of these date calculations must be passed back to the FR’s case management software so that they have a central place where they can reference when and where they need to make their next visits.

3. Project Gemini and CE Proof of Concept

The basic methodology of CEQ and CED has remained the same since 1980, with the biggest change being the conversion from PAPI to CAPI but still using similar sections and questions. With rising costs to conduct a survey and a desire to reduce respondent burden, lower measurement error rates, and leverage the latest technologies that are available, BLS is exploring ways to update the CE instruments so interviews are conducted more efficiently without a negative impact on data quality. They have decided that a major re-design of the CE surveys is in order. They have an initiative known as the Gemini Project to research proposed changes to CE and develop what will be the next generation of the CE instruments. At this stage in the Gemini Project, BLS is working with the Census Bureau to develop a new instrument that combines CEQ and CED into one single instrument that is split into two interviews to collect CEQ information, a Records interview, and a Recall interview. Since both CE instruments are already in Blaise, theoretically they can be combined together using common code blocks, but there will still need to be adjustments made to this combined instrument. There is also a web-based instrument/app that is being incorporated into this survey, which will be used to collect diary data for individuals in the CU (rather than a single diary for the whole CU) over a one-week period. The idea is that respondents should access a mobile site or an app on their phone or tablet at the time they are making their purchase during the diary period, in order to obtain more detailed and accurate diary data. Since this new instrument for the Gemini Project is in the early stages, it is known as the CE Proof of Concept, CE POC, or simply POC.

CEQ and CED are very different in their approaches to collect expenditures, but they do have some overlap in the information they collect. Both instruments use similar household demographic sections, and they ask the respondents about their income, if they have a mortgage, how many cars they own, and their spending on groceries. However, CEQ asks more detailed questions about expenditures. In considering the design of the Proof of Concept instrument, we decided to use the current CEQ instrument as the baseline and add the appropriate CED sections as needed. The sponsors reduced the number of CEQ sections from 22 to 16 and split the collection of the sections into two separate interviews. The first set of CEQ sections make up the Recall interview since they ask about expenditures that a respondent is likely to remember, such as what they spent on a vacation or on clothing. The second set of interview sections are the Records interview because the CU is likely to have records of those expenditures, such as mortgage or rent payments. What the POC instrument did get from CED were the sections that related to placing and picking up the diary, including the “recall” table to add additional expenditures that the CU did not enter in their diary but remembered during the second visit. Besides the CED sections, other code would need to be added to make the POC fit the requirements that the sponsors wanted.

4. Outcome Codes (Control Codes)

All CAPI surveys conducted by the U.S. Census Bureau have a set of outcome codes (control codes) that are assigned to each case. These codes are status codes that tell our Laptop Case Management (LCM) software to perform an action on the case and allow Headquarters staff to see the status of the case. The codes are also an easy way to reference what happened with a case, such as the reason a case was a non-interview. CEQ is very straightforward with its outcome codes in that they are set as an overall code for the entire case. However, CED not only has an outcome code for the entire case, there are separate outcome codes set for placing the diaries or picking up the diaries. This is to account for all potential possibilities that can occur during each visit, such as a respondent who filled out a diary one week but did not fill it out another week, a diary being refused, a respondent who is out of town during the first week, or both diaries being completed. Since POC involves diaries, we decided to use the same LCM interface that CED uses. The POC needed to set the same diary placement and pickup codes along with overall outcome codes for a case. There were also a few new outcome codes added to track whether or not the Recall or Records interviews were completed.

The sponsors had to consider all possible results for a case, and it meant making sure the instrument could handle each result and set the appropriate outcome codes. This was important because some outcome codes determine the flow of the interview. For example, if a non-interview outcome code is set, the instrument will skip over all middle sections and go straight to the back. There is also an FR debriefing section that only comes on-path when the Recall Interview is complete and the diary has been placed with the CU (or it was refused, or the interview was completed after the interview date window).

In the following example, the case on its second visit (VISITNUM) and the diary pickup (Outcome_Pickup) and Recall Interview (Outcome_Recall) were completed. However, the Records Interview is not complete (Outcome_Records = 202) so the case is given an overall outcome of 383. Each Recall or Records section has a flag that tells the instrument if that section is complete during the current interview. Once all of the sections are completed for their interview, the appropriate outcome code is set and that code will change the overall case outcome code.

```
IF SectComp[2] = YES AND SectComp[3] = YES AND SectComp[4] = YES AND SectComp[10] =
YES AND SectComp[11] = YES AND SectComp[13] = YES AND SectComp[14] = YES AND
SectComp[15] = YES AND SectComp[21] = YES AND SectComp[22] = YES THEN
    Outcome_Record := '201'
ELSEIF SectComp[2] = YES OR SectComp[3] = YES OR SectComp[4] = YES OR SectComp[10] =
YES OR SectComp[11] = YES OR SectComp[13] = YES OR SectComp[14] = YES OR SectComp[15]
= YES OR SectComp[21] = YES OR SECTCOMP[22] = YES then
    Outcome_Record := '202'
ELSE
    Outcome_Record := EMPTY
ENDIF

IF VISITNUM = 2 AND Outcome_Recall = '201' AND (
(Outcome_Pickup <> '201' AND Outcome_pickup <> '325' AND Outcome_pickup <> '326' AND
Outcome_Record <> '201') OR
((Outcome_Pickup = '201' OR Outcome_pickup = '325' OR Outcome_pickup = '326' ) AND
Outcome_Record <> '201') OR
(Outcome_Pickup <> '201' AND Outcome_pickup <> '325' AND Outcome_pickup <> '326' AND
Outcome_Record = '201'))AND Admin.ReplaceSpawn <> Yes THEN
    Outcome := '383'    {Visit 2 follow-up (pickup or records incomplete)}
ENDIF
```

Fig. 1 – Display of all outcome codes that are set in the POC. “Records Outcome” is not a completed outcome code therefore the current overall case outcome code is set to 383, “Follow-up Required (Visit 2)”

4.1 Outcome Codes Everywhere

With CE being one of the first surveys written in Blaise, it meant that there was a period of adjusting our thinking about how to program instruments in Blaise vs the previous authoring language, CASES. One example of this is how the outcome codes were set. The normal approach for setting outcome codes in CASES was to set the outcome code immediately after reaching the condition in the survey that satisfies the outcome. So setting each outcome code was placed in-line in the instrument code during the execution of the survey, meaning outcome codes were being set throughout various sections. With Blaise's rules re-execution engine, that isn't necessary since the rules are checked all the time. Therefore, an instrument can have a separate block or procedure that will run at the end of the rules in the datamodel. However, the original implementation of CEQ and CED used some CASES style methodology in there. While the setting of outcome codes worked, it also presented difficulties when outcome codes needed to be changed since multiple sections of code had to be checked and verified before the change could be implemented. Despite the maintenance challenge, the outcome code assignments were left in place because they were working and there was less of a desire to risk causing a problem in production.

4.2 Outcome Code Procedure

Since we were making some major changes to CEQ's code to accommodate CED and we had to add the CED placement and pickup outcome codes, we decided that we would move setting all of the outcome codes in the POC instrument into a procedure. This meant finding every single outcome code assignment and moving them into a single procedure that would be called in the back of the instrument. Within the new outcome code procedure, we would write the appropriate universe statement for each outcome code assignment and thoroughly test the procedure to make sure all of the expected outcomes were reached. This included not only the overall case outcome codes, but the diary placement and pickup outcome codes

as well. Since CE has a well-established set of testing scenarios and we implemented this change early in the project, we were able to conduct a very thorough internal test before the instrument went out for system interface testing. We immediately saw the benefit of using the single procedure as we were able to easily test and examine why an outcome code was or was not what we expected. It was also beneficial to have the expected conditions for assigning that code right there in the universe statement. We did not have to search our code to determine if an outcome code assignment was buried inside a nested universe in some other section of code. This approach will make future changes to the outcome code easier as only one section of instrument code has to be tracked and maintained instead of several sections of code.

5. Recall and Records Interview Sections

With the CEQ sections being split into the Recall (Visit 1) and Records (Visit 2) interviews, that meant shuffling the sections so that they are asked in the correct order. For example, the sponsors decided to ask Sections 8, 6, 17, 18, and 19 (in that order) for the Recall interview and the rest of the sections would be asked in the Records interview. Since each CEQ section is an individual block in Blaise that is called from the main datamodel, setting up the Recall and Records sections were a matter of moving the calls to the sections to the appropriate places in the rules of the datamodel. Additionally we added a universe to check and see if the value of VISITNUM was 1 or 2 so that the appropriate sections would be put on path.

The sections are declared as parallel blocks so an FR can complete the sections in a different order if they want to. Having parallel blocks was also helpful when a new requirement was added to have the instrument not ask the Recall sections again during the Records interview, but still allow the FR to have access to the Recall sections. If for some reason the CU forgot something in the previous visit, the FR can still access those sections via parallel tabs and add that information. However, the questions are not on the main instrument path during the Records interview so the FR does not have to press Enter multiple times or the End key to get past those sections. Blaise has been flexible in allowing the instrument to keep the proper sections on path without losing data or making the FRs repeat sections that are not required for that interview.

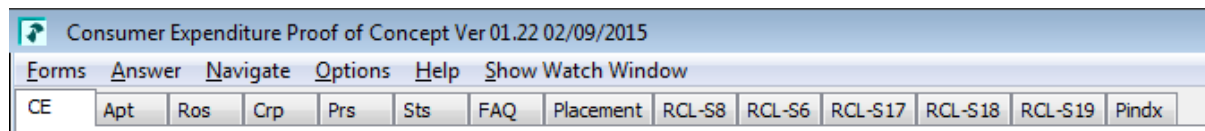


Fig. 2 – All available sections in the Recall Interview

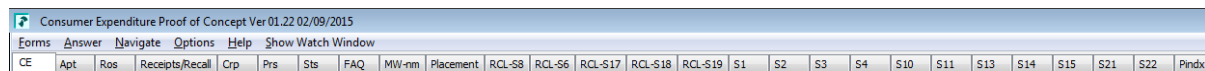


Fig. 3 – All available sections in the Records Interview

Many more sections are put on path in the Records interview. Besides the Records sections, there is also a section to show the list of usernames assigned to the members of the CU if they opted for a web diary. The Receipts/Recall section is also on path for the FR to enter any expenditures that were not recorded in a diary but were recalled by the CU during the interview. The Recall sections are still available in case any additional expenditures need to be added during the Records interview.

6. Tracking Diaries

Another challenge in programming the POC instrument was how to properly track the diaries. In a regular CED interview, the FR leaves a paper diary that is filled out by the CU. When the FR returns and picks up

the diary, they send it back to their regional office (RO). The RO takes that diary, assigns it a barcode, and it is shipped to the National Processing Center (NPC) in Jeffersonville, Indiana for scanning and processing. With the Gemini Project, there is a requirement to move away from using paper diaries as the means of collecting expenditures for CED. In 2013 a test was done with a sub-sample of CED respondents that gave them the option to use a paper diary or to fill out the diary on the Internet using a provided username and password. For that test, each CAPI case was pre-assigned a username and password that the FR would give to the respondent if they opted to fill out a “Web Diary.” In 2014, a different test was conducted on a sub-sample of CED respondents, known as the “Individual Diary” test. Each eligible member of the CU was given the option to fill out a Web Diary using their computer or a mobile device (an eligible member of the CU is a member over the age of 16 who is currently living at the household). Instead of one diary for the entire CU, there was a diary for each eligible individual of the CU who opted in to filling out the diary. For that test, no paper diaries were used for the sub-sample. If they wanted a paper diary, they were screened out.

The methodologies from both of these tests are being combined for the diary component of the POC. The goal of this new Web/App Diary is to allow respondents to record their expenditures at the time of purchase using an app on their phone or tablet in order to capture more accurate and complete diary data. After the Recall Interview is completed, each eligible member of the CU is asked if they regularly access the Internet. If they do, they are given the option to fill out a Web/App Diary. A member who opts for the Web/App Diary is given a pre-assigned username and password so they can access the diary throughout the collection period. Due to security requirements, the password is required to be long and complex. This raises usability concerns as the respondent is required to key in this potentially challenging password every time they open the app. However, a member of the CU may instead choose to fill out a paper diary or they may decide they do not want to participate in recording their expenditures. The instrument will capture how many paper diaries were given out so that the RO and NPC know how many paper diaries to expect for processing.

6.1 Web/App Diary Logins

Being able to provide diaries to individual members of the CU and have them fill them out on paper or on the web introduces a number of possibilities and requirements that must be considered for the POC instrument and for survey procedures. A big requirement to consider is security and being able to have enough web diaries for all members of the CU. The POC’s household roster allows for up to thirty members in a household, and theoretically they could all be part of the same CU. The Centurion software that is used for the Web/App Diary does not allow for custom accounts to be created. Even if it did, there would need to be a way to connect that web diary with that POC case. It is also possible that an FR may be conducting the interview in an area where they may not be able to access the Internet to allow for account creation. Even if there is Internet access available, there is a question of making sure the connections and any data being transmitted is secure.

In considering some of these concerns, the sponsors decided that the POC instrument would display the web diary accounts needed based on the household roster, and that the accounts themselves would be pre-created in Centurion. Since each POC case is a single Blaise database, we stored all of the potential usernames and passwords needed into one block. Once the FRs collect the household roster information, the instrument determines the appropriate CUs. If a CU member opts for the web diary, the instrument will display a username and password that is given to the member to access their web diary. For security

purposes, after the web diary username and password is provided to the CU member(s) on the first visit, the FR will not be able to access information about the passwords from the instrument. Instead, we added a screen in the instrument to let the FR see a web diary username only, in case a CU member forgets that and calls the FR. There is a help desk that a CU member can call if they need to have their password reset. Otherwise, once the username and password is given to the CU member, accessing the diary and filling it out is completely their responsibility.

Consumer Expenditure Proof of Concept Ver 01.22.02/09/2015

Forms Answer Navigate Options Help Show Watch Window

CE Apt Ros Crp Pys Sta FAQ Placement RCL-S8 RCL-S6 RCL-S17 RCL-S18 RCL-S19 Pindx

♦ Instruct the respondent and all available household members on how to use the paper and/or electronic diary(ies) and provide them with user guide(s).

For electronic diaries: Be sure to go over with the respondent:

- how to log in with their username and password (attempt to log in with a respondent, if they are willing)
- creating shortcuts to the web and mobile diaries
- show the respondent(s) the link to the video tutorials in the User Guide and inform them that the links also appear in the electronic diary

Transcribe the Individual Diary username(s) onto the User Guide(s) and password(s) onto appointment card(s) for the respondent(s).

For paper diaries:

Instruct the respondent(s) on how to complete the paper Diary leave the Diary form(s) with the respondent.
Transcribe the control number onto the paper diary. Control Number: 0406013GD3500010110100

Resp.	LN	Name	Diary Type	Username	Password
XXX	1	John Citizen	Mobile/Web	web01111	11111111
	2	Jane Citizen	Mobile/Web	web22222	22222222

1. Enter 1 to Continue

Web Information

Web Information

Reminder

00000002 USER NAMES 12:14:42 AM 2-23-2015 VISIT NUMBER: 1 RESPONDENT NAME: John Citizen 2506/2524

Fig. 4 – Screen where the FR can provide login information to the CU members at the time of diary placement

6.1.1 Spawned CUs

As we were programming the POC instrument, we realized there was a possible situation that could arise with a case and its handling of the usernames and passwords. As mentioned in the design of the original CE instruments, a CU is a group of people living in the household and typically there is one CU per household. However, there are households where you might have multiple CUs, due to living situations such as roommates or multiple families who each contribute their own money to household expenses and make expenditures exclusively for themselves. Both CED and CEQ have sections that ask if certain members of a household (such as non-relatives) pay for their own expenses with their own money. If the instrument determines that a household member is a part of a different CU, the instrument and the LCM software work together to create a separate case for the FR to interview the other CU. To accomplish this task, we run a Manipula script after the FR exits the case. The script will generate a new Blaise database and populate it with information from the original case. LCM handles adding the newly spawned case to their database so that it is properly added to the FR's assignments and is transmittable back to headquarters when it is completed.

Since the instrument can spawn new cases to accommodate multiple CUs, the question became how to handle the new case so that a spawned CU would also have access to a Web Diary to fill out their information. Since the Web Diary accounts have already been made, there currently is not a way to create

them on the fly. If that was possible, we could use Blaise to calculate a new username and password for the FR to give to the CU members who opt for a Web Diary. Since that was not an option, we decided to copy over some of the unused usernames and passwords from the parent case to the spawned case for the members of the new CU so that they would have access to the Web Diary. Since each username and password is connected to one line of the roster, it is a matter of copying over the appropriate username and password via Manipula when the spawn case is created. However, this approach means that any household members added to the roster in the spawned case will not have a username and password available for them. This should be a very rare occurrence since their name and status as a household member should have already been collected in the parent case.

As an example of how we made this work, we could have two couples living in the household that each pay their own expenses. Since that would create two CU's, Person 1 and Person 2 on the roster are their own CU and stay with the parent case. Person 3 and Person 4 will be interviewed in the spawned case. The username and password that would be available for Person 3 and Person 4 in the parent case is copied over into the spawned case via Manipula. We have found in our own systems testing that this does work without an issue.

7. Using the Web/App Diary

When a user logs into the Web/App Diary for the first time, they will enter the date they were told to start recording their expenditures. After that they are taken to a screen to enter their expenditures.

7.1 Internet Version of the Web/App Diary

Fig. 5 – The main screen of the Web/App Diary

Selecting a category will bring up additional information to be entered about their expenditure.

The screenshot shows the 'Add an Expense' form with the following fields and values:

- Date:** February 8, 2014
- Description:** Sandwich
- Cost:** \$5.12
- Category:** Food and Drink Away from Home
- Meal Type:** Select One
- Alcohol Included?** (Select all that apply):
 - ☐ Wine
 - ☐ Beer
 - ☐ Other
 - ☐ None
- Cost of Alcohol:** \$0.00

Buttons: Clear, Save

Fig. 6 – Follow up information about “Food and Drink Away from Home” purchases.

The screenshot shows the 'Add an Expense' form with the following fields and values:

- Date:** February 9, 2014
- Description:** T-shirts
- Cost:** \$9.58
- Category:** Clothing, Shoes, Jewelry, and Accessories
- Gender:** Male
- Age:** 16 & Over
- ☐ Purchased for someone outside your household

Buttons: Clear, Save

Fig. 7 – Follow up information about “Clothing, Shoes, Jewelry, and Accessories” purchases.

As expenditures are added, they are displayed under the summary of expenses. The user is also able to edit or delete expenditures.

The screenshot shows the 'Summary of Expenses' section with a table of entries:

Date	Description	Cost	Category
February 12	Gas	\$26.89	All Other Products and Services
February 9, 2014	T-Shirts	\$9.58	Clothing, Shoes, Jewelry, and Accessories
February 8	Coca-Cola	\$1.75	Food and Drink for Home Consumption
February 8	Sandwich	\$5.12	Food and Drink Away from Home

Buttons: Edit, Delete (X), Cancel, Save

Fig. 8 – Editing a Web/App diary entry.

7.2 Mobile Device Version of the Web/App Diary

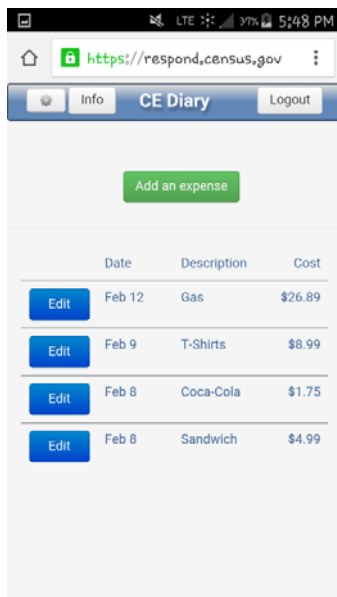


Fig. 9 – App Main Screen

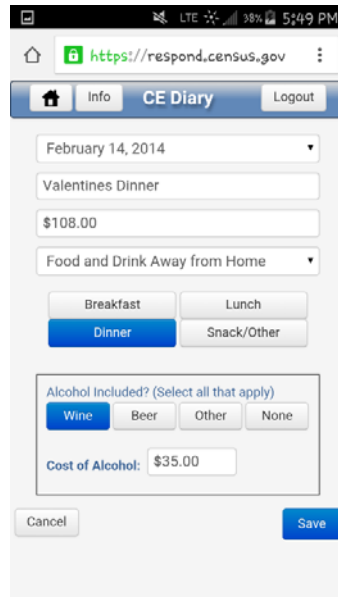


Fig. 10 – Adding a new expense in the App

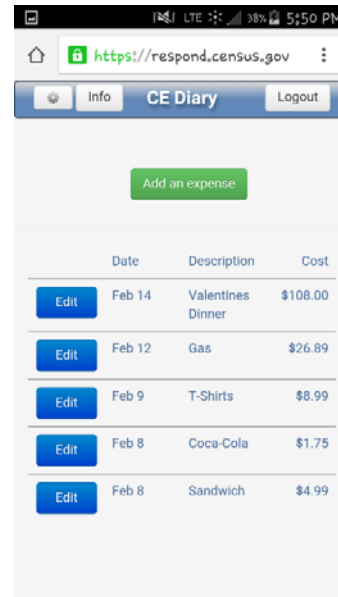


Fig. 11 – After adding an expense

8. Future Developments

The long-term goal of the Gemini Project is to create a system that more efficiently and accurately collects the data required for the CE survey. To accomplish this goal, the Census Bureau hopes to use a combination of data collection methods that seamlessly interfaces with their control and tracking systems so that the staff at Headquarters can examine the status of each case and its individual diaries in real-time. This summer, we will be fully testing the POC instrument using the combination of the Blaise instrument to conduct a majority of the CE interview and the Web/App Diary (developed in Centurion) to collect the individual diary information. The results of the test and the lessons learned from programming this instrument will be analyzed and used to enhance the overall design and implementation of the redesigned CE survey.

9. Conclusion

For a large and complex survey like CEQ or a smaller and date-sensitive instrument like CED, Blaise has handled the requirements from BLS and from the Census Bureau with success in collecting crucial economic data. However, Blaise does not have to do everything all by itself. By leveraging other available technologies that provide other approaches, we can help streamline the data collection process. If we can use them to interact with the Blaise instruments and data, it opens up further opportunities to tap into the power of Blaise and what it can do with our data. The POC has already provided new ideas for us to consider applying to other surveys in the future.

10. Acknowledgements

The author would like to acknowledge the work and knowledge of Latha Srinivasamohan, Michael Johnson, and Jacob Tomlinson as their assistance on the CE Proof of Concept instrument was invaluable in meeting the instrument requirements and overcoming the challenges that were presented.

The views expressed in this paper are those of the authors and not necessarily those of the U.S. Census Bureau.