

CONTINUING THE BLAISE 5 JOURNEY MIXED MODES AND BLAISE 5 INTEGRATED EXTENSIONS

Authors: Angela Belo, Alessio Fiacco, Nafiis Boodhumeah, Sven Sjodin

At the National Centre for Social Research, we have been continuing our journey to migrate more surveys to Blaise 5 using various combinations of modes: CAWI only, CAWI and CATI, CAWI and CAPI. We have also extended the functionality of Blaise 5 to integrate with other systems for operational support of our surveys, particularly our Survey Control System (SCS) and Case Management System (CMS). For some projects, we also integrated with other third-party apps for added functionality to the survey, such as interactive map component and calendar of life events.

We would like to share our experiences, what we have developed, what we have learned and what is yet to come. We will primarily use the Wales National Travel Survey (WNTS) as an example to highlight the various elements that we experienced. We will also reference the English Longitudinal Study of Ageing, Life History survey (ELSA LH) and the Technical Education Learners Study for some specific experiences.

1 Background and Overview

The WNTS survey was commissioned by Transport for Wales (TfW), which is a Welsh Government-owned company. It focuses on travel behaviour and opinions within Wales across a range of transport modes. The sample was drawn from the postcode address file, including 14,900 addresses for 2025-26. The survey is bilingual (English and Welsh). The web survey can be completed in Welsh, and a Welsh interview can be requested for face-to-face interviews.

This is a web-first survey. If there has been no web participation, then an interviewer selects a single member of a household to take part face-to-face (F2F). Respondents are sent an invitation to take part online: any two adults can take part online but only a single member of the household is selected to take part during face-to-face interviewing. This is due to difficulties in enforcing selection procedures online.

The survey consists of a F2F CAPI interview with a self-completion (CASI) section. A telephone interview (CAPI-Tel) option is also available for those who request it. The two-day travel diary includes a map function that respondents can use to select their origin, and destination points for each trip. This function requires access to the internet; the interviewer will need to hotspot the laptop to their interviewer mobile device. For web respondents the survey includes the two-day travel diary with the map component all integrated together.

Respondents who complete a productive interview are entitled to a digital gift voucher (for web respondents) or a physical gift card (for CAPI interviews).

2 Various Mixed and Multi-Modes

2.1 CAWI and CAPI mixed mode using separate instruments and databases

The WNTS survey is set up as two separate projects with separate Blaise instruments and databases. The structure and design of the two instruments are identical, or near identical so that information can be updated between them during the overlap of CAWI and CAPI fieldwork.

Manipula scripts are used to update information from one database to another (web completes, opt-outs, and contact updates). Prior to Capi fieldwork beginning, these updates will prevent cases from being pushed out to CAPI interviewers. During CAPI fieldwork, interviewers are notified via email and SMS text message of any cases which have completed online or opted out that are allocated to them, with instructions for coding them out.

It should be noted that CAPI interviews are conducted offline, disconnected from the server, while CATI or CAWI surveys are completed online, directly updating the server database. Also, once CAPI interviewers download their allocated cases, any further updates to those cases captured on the server are not downloaded again. CAPI interviewers are instructed to connect and synchronise their device to the office daily; to upload any work they have completed, download any additional projects / cases they were allocated, and receive any email notifications. Sometimes the delay in reconnecting can be longer than a day. This presents challenges in keeping information updated between the CAWI and CAPI database, to be certain which record is in fact the most up to date.

Besides the Manipula scripts to keep information updated between the two databases throughout fieldwork, there is also a need to reconcile differences between the mode databases afterwards.

2.1.1 Reasons for Separate Databases for Web and CAPI

We currently do not have a solution for a single database for all modes, which would need to include Write Interceptors to control the updates to the master or historical database. These write-interceptors are Manipula setups that are installed and executed on a Data Server. NatCen's business decision is still outstanding on this matter. There is a desire to move to a single database solution, however further work is required to define a comprehensive set of rules to control the updates before the solution can be implemented.

Among the considerations there are, for example, the rules especially for dealing with updates coming from various sources, i.e. CAPI interviewers working offline (with possible delays), office updates including opt-outs/refusals, allocation of cases to CAPI interviewers and reallocations / re-issues to other CAPI interviewers, respondent contact updates, CATI / CAWI completed or partially completed cases, and updates from the Session database.

As stated previously, Web and CAPI are open at the same time and CAPI interviews are done offline. Moreover, data may need to be merged not only from different modes, but potentially different sources (main, session or history database).

There needs to be a process to copy both Web and CAPI data to some sort of backup or merge history database and then select the appropriate cases according to a comprehensive set of rules. At the University of Michigan, they have implemented Write Interceptors, and their rules are very complex.

2.2 CAWI and CATI mixed mode using single database and launcher

Since 2022 we carried out a few CAWI and CATI mixed mode surveys in Blaise 5. Initially, we used a single main survey script with the CATI administration block included (using the INHERIT CATI statement). This is the approach we adopted for our first Blaise 5 project, the Technical Education Learners Survey 2022.

For the subsequent waves of the survey, we switched from the Traditional Approach (Dial, Appointment and Main data models) and adopted the Launcher Approach (also known as, the multi-topic).

With the help of the Blaise Team, we changed our original Dial questionnaire into a Launcher.

Initially we struggled to understand the Multi Topic CATI example, especially in terms of passing information to and from the Topic as well as dealing with a multi-topic project, with a Login instrument and unable to start the survey from the Launcher.

2.2.1 Rationale for the Change

Using the Launcher is a way to separate the real topic-specific questionnaire data from the administrative data relating to scheduling or calling respondents.

With multi-topic projects all CATI-specific information will be stored in the Launcher such that the Topic instruments remain CATI-agnostic in that they don't contain any CATI-specific fields. This makes it easier for multimode when the same questionnaire is also used for CAPI and CAWI modes. The Topic instrument does have a CATI-specification file (.btx), however the data model itself only contains fields for the actual questions of the survey.

The main reason for using the multi-topic approach was because it was seen as similar to the CAPI approach, which uses the CMA Launcher, and that it would be more consistent across modes going forward. In the next releases of Blaise, the Multi-Mode Launcher for central administration of cases, would sit on top of the CAWI, CAPI (CMA) and CATI Launchers.

Also, the Blaise Team suggested that if we standardise the Launcher, it could be used across all or several projects. So far, we haven't been able to do so but hope to in the future.

2.2.2 CAWI and CATI Considerations

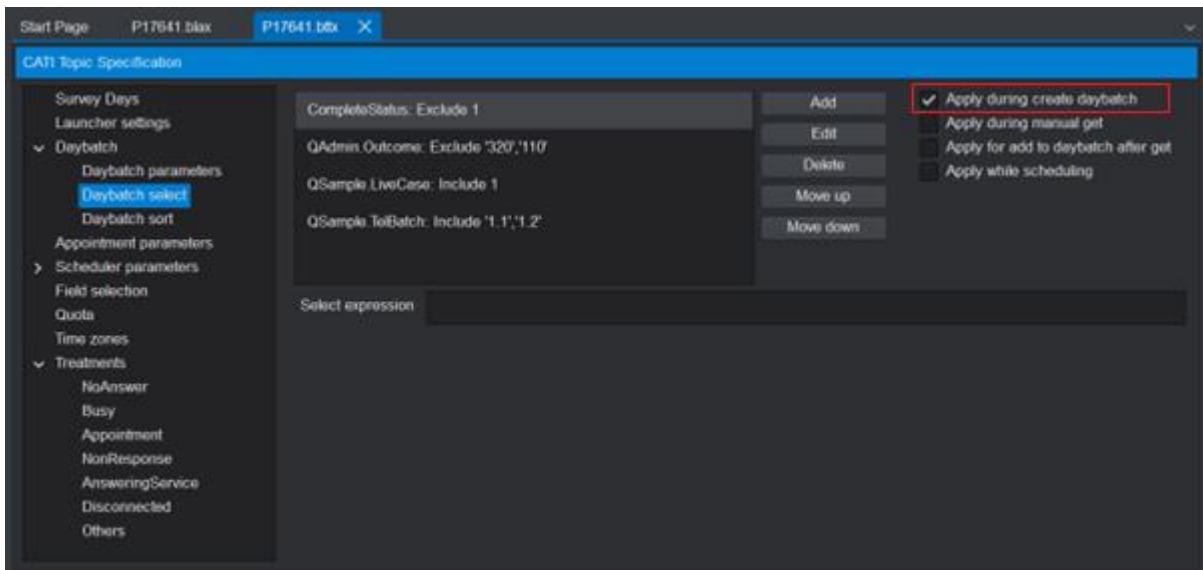
It was important to be able to control the cases going into the DayBatch, as these are the respondents that the Telephone Interviewers would then contact.

We used a DayBatch select filter, to exclude CAWI completed cases and cases where the respondent contacted our Survey Enquiry Team (i.e. Help Line) to opt-out of the survey We needed to ensure that these respondents were not contact again.

The system used by the Survey Enquiry Team to record respondents opting out, sets the NonResponse dial result, so those cases are excluded during the creation of the DayBatch. Also, the Blaise system changes the status of cases completed in other modes in the DayBatch. The status of a case is set to "NoNeedToday" and the scheduler no longer attempts a dial:

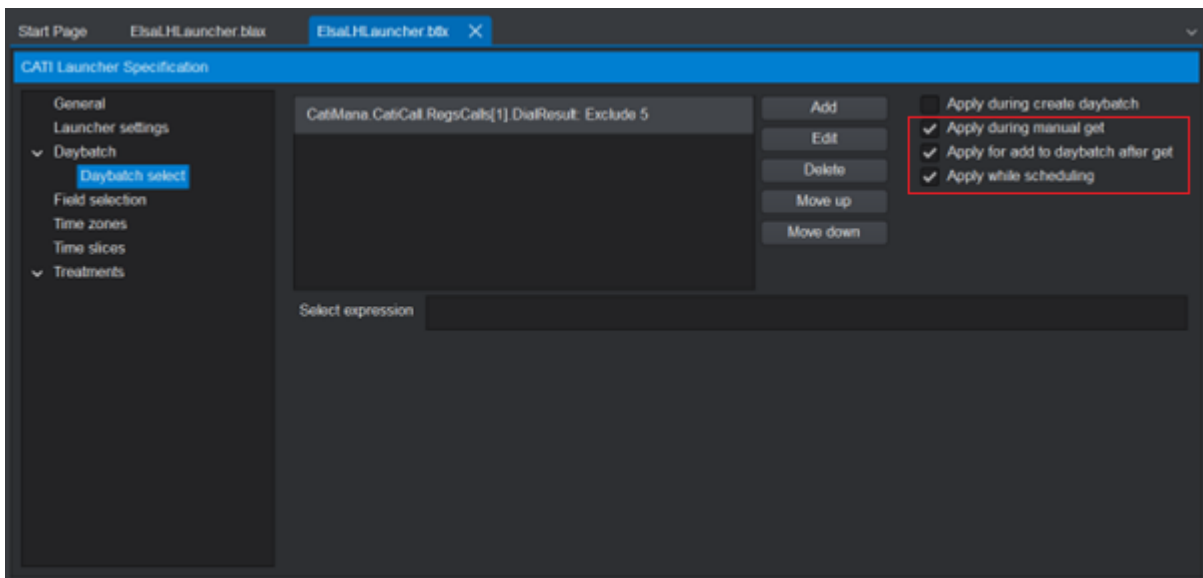
Set Form Status for key '999999001' from 'Default' to 'NoNeedToday'. Reason: case completed in another mode (CASI)
--

The DayBatch select filter was applied while scheduling and during the creation of the DayBatch to exclude completed cases (CompleteStatus=1) and certain outcomes, derived in the Web questionnaire if the respondent failed an eligibility check.



In addition, since our interviewers could manually select a serial number (on our CATI receipt page), there was potential for the interviewer to contact a case that either had become completed in CASI mode or where the respondent had opted out of being contacted about the survey. We needed to prevent this from happening.

Recently, we added to the Launcher a DayBatch select filter to exclude cases where **CatiMana.CatiCall.RegsCalls[1].DialResult = (5) Nonresponse**. We selected the 'Apply during manual get' option to ensure these cases didn't get added manually.



Unfortunately, this additional select filter had a performance penalty. When delivering a case, the system carries out an additional check at that 'Select Field' and then excludes the case as appropriate.

Therefore, with the version of Blaise used in Production during fieldwork (5.13), creating the DayBatch had an impact on the available memory of the server. The latest version of Blaise comes with performance improvements in this area.

2.2.3 Differences in Recent Projects

Resource Database

After the last wave of TechEd in 2024 and in advance of the ELSA Life History study we created a single resource database for all instruments in the multi-topic solution. This included the Login instrument, which is used by Web respondents who enter their case with an AccessCode. The Main Topic questionnaire uses a numeric primary key, so a Login instrument is required for Web and CATI projects so that Web respondents can use an AccessCode, for greater security.

This change was made to ensure that only one BLRD required maintenance, but also to avoid CSS-related error messages due to “inheritance”. When starting with a Login instrument that redirects (with the StartSurvey action) to the main questionnaire, if the CSS-filenames in the survey root of all chained project aren’t the same, it causes layout problems.

Example: The Login instruments uses:

Web_Large-ENG.css
Web_Small-ENG.css

The Topic instrument uses:

NatCen-Large-Default.css
NatCen-Small-Default.css

The CSS-filenames are a combination of the LayoutSet and Languages, and they must be the same in all the chained surveys. We standardised the LayoutSet names so that they are the same across all surveys.

Session Data

In the first CAWI and CATI project we did, back in 2022 (with an old version of Blaise 5), we had some issues with cases in the Session database (in a multi-topic instrument), which affected the creation of the CATI DayBatches. In a nutshell, completed CAWI cases went into the DayBatch because the case existed in the Sessions database (the session data for some completed records hadn’t been removed from the Session Database). Hence, we applied the setting to delete Session Data at Session time out and on quit and used that setting ever since.

In ELSA we reversed that decision. During the Pilot we had experienced some data loss in a small number of records. The Session Data for unfinished records was useful to recover data in Blaise. So, by not deleting the session record at least we had a backup.

The Session Data turned out to be a bit of a double-edge sword during ELSA’s fieldwork. With the version of Blaise used we weren’t able to update the Session Data in ELSA (via Manipula). We also became aware that when the Survey Control System (SCS) would carry out some sample updates (e.g. address or email) or outcome updates via the API and synch them to Blaise, these changes would be overwritten by the Session Data of unfinished records.

Record Locking (in the Launcher)

After TechEd 2024 we discovered the need to set ApplyRecordLocking="false" in StrictCati for the Launcher. We made this change as interviewers were receiving the "Questionnaire is locked" error in the Launcher.

We hadn’t experienced this problem previously, but the Blaise Team’s general advice was to disable record locking for the Launcher data model and only use this setting for the Topic questionnaire.

3 Integrating Systems and Components with Blaise 5

3.1 Redesign of Sample Management / Survey Control System (SCS)

Early on, we recognized the need to replace and redesign our Sample Management System (SMS) to integrate with Blaise 5 and to work with various multi-mode surveys. The current sample management system (SMS) is used with CAPI surveys in Blaise 4 predominantly, with a few exceptions. So, the functionality needed to be extended.

The Survey Control System (SCS) has been developed over the past 2-3 years, with a tight integration with Blaise 5 using API:s to synchronise sample data from SCS to Blaise, push allocations information to Blaise and our Case Management System (CMS) through the CMA Launcher (provided by the Blaise product), and import Blaise administrative and communications data to SCS.

The SCS system includes features to manage the operational aspects of conducting surveys and below is a summary of these key features.

In SCS, there is functionality to define all aspects of a project, including links to the Blaise instruments, waves and modes involved, definition of the sample structure and imported data structure (administrative and communications blocks) mapping to Blaise. Sample data is loaded into SCS and then synchronized to Blaise. As surveys are conducted and data is captured in Blaise, a subset of it is imported from Blaise into SCS, on an hourly basis.

The SCS has the ability to generate automated communications to respondents, such as invitations to participate, reminders to complete the survey, and thank you for responding, either by email, SMS or letter.

There are features to setup the automatic processing and sending of digital incentives (gift vouchers) based on defined criteria for the project. This feature works from the completed survey data imported from Blaise into SCS and also integrates with our gift card / voucher provider, Love to Shop, to generate digital vouchers and our email provider, Spotler MailPro, to send digital voucher information to respondents.

The Freephone (Help Line) functionality allows the recording, tracking and resolution of issues from respondents. For certain log types, it updates Outcome codes in SCS as well as directly into Blaise.

The SCS has the ability to record and merge contact updates from multiple sources, which can then be used in subsequent communications to respondents.

There are features to support the automatic and manual allocation of cases to interviewers, issuing to mode, and reallocation / re-issue of cases to different interviewers, as and when needed. These features work from the uploaded sample information in SCS, and the known CAPI interviewers, their location and availability. Once allocations have been approved and issued to mode, the cases are pushed to the CMA Launcher ready for downloading by the CAPI interviewers.

The system handles automatic generation of interviewer notifications of case changes once they have been downloaded to work offline.

3.2 Integration with Case Management System (CMS) using the CMA Sync.

We have our own Case Management system on the interviewer devices, which had to be made able to run Blaise 5 surveys. It uses an MS Access database to store data about the surveys, waves and cases. This needs to be populated when new surveys and cases are downloaded and updated after data entry sessions. The Blaise Team has enabled us to tap into the synchronisation functionality of CMA via a Manipula setup that calls CMA with the applicable credentials and connection details.

For this to work, the SCS populates the CMA Launcher database as well as the Blaise survey data on the server when cases are issued to interviewers. We use the CMA_CustomUse field in the Launcher to provide the field names and values to update the CMS database on the devices. Due to a 32-bit restriction of the MS Access driver, we use a separate Manipula setup to populate the CMS database. This reads from intermediate SQLite databases populated by the synchronisation setup.

The data entry sessions are launched by a DEP.exe call from CMS and the updating of case data in the CMS database after the session is handled by using the Blaise 5 API.

3.3 Integration with Third-Party Apps / Components

3.3.1 Map Component

The WNTS survey has the map component integrated into the Blaise questionnaire.

The map itself works similarly to Google maps but is based on Ordnance Survey. The map accepts both the manual searching of addresses through address lines or postcodes and allows respondents to drop a pin onto the map to select a location. Once a location is selected, the location will be shown as a pin on the map with a label i.e. 'County Park'. The respondent will then be asked a series of questions about this journey, such as what form of transport(s) they used and the reason for the trip.

Statistics Netherlands / Team Blaise helped develop this function and the integration of it with our Blaise 5 survey.

3.3.2 External Lookup to Validate Gift Cards

In Blaise 5, external files (such as IncentiveCardNos) operate differently compared to Blaise 4.

The 'IncentiveCardNos.csv' lookup file is located in "C:\WinCMS\Incentive" and is used for all Blaise surveys. WNTS is the first Blaise 5 survey to utilise this file.

For Blaise 5 packages, the main points to bear in mind when working with externals are the following:

You cannot rely on a folder that is not managed by Blaise (such as C:\WinCMS\Incentive). This cannot work, for example, on Android devices through the Android App, as the folder cannot be installed or accessed by the App.

For Blaise, the package (.bpkg) must be self-contained and include all necessary files to run a questionnaire. This involves copying the version of incentivecardnos.csv to the deploy folder to ensure the latest version is used.

Most external files, which remain static during fieldwork, can be included in the package.

We had to come up with a workaround for incentive cards, but in the long term this may require further development. In the short term, the script generating 'incentivecardnos.csv' will be modified to also copy this file to the survey deploy folder.

Although there are no upcoming Blaise 5 CAPI surveys involving gift cards, a permanent solution will be required. This will likely involve the case management system copying the file to the deploy folder before calling DEP.exe.

3.3.3 Interactive Events History Calendar (EHC)

For the English Longitudinal Study of Ageing Life History (ELSA LH) survey there was a requirement to include an interactive calendar of life events. This was to assist respondents in the recall of life events and timings.

ELSA Life History was aimed at adults aged 50+ and their partners, covering various topics including children, partners, places respondents have lived and worked and their health.

Evidence suggests that EHCs can help respondents recall and organise events from their past, however many of these studies were conducted using face-to-face interviews. This calendar was designed to be used as a tool for web respondents and is one of the first to test the effectiveness of an EHC in a web survey. We were also interested to know if this calendar could be used in the CATI element of this project to help respondents recall events from their past. As a result, telephone interviewers were asked to refer to the EHC when interviewing respondents over the phone. CATI interviewers found the Calendar helpful in the pilot and hence it was also used in the mainstage.

Key features of the calendar component include the following.

- EHC spans respondent's entire life, from birth to today
- Displays age for each year; with scroll bar to see more years.
- As questions are answered about life events (e.g. children, partners) these will appear in EHC.
- Each category of event has a different colour.
- A summary row for each category shows all events within that category (e.g. all their children)
- Individual event rows are shown underneath (e.g. name of each child).

There are provision for editing an event or period, if entered incorrectly.

The ELSA Life History is a CAWI and CATI survey with the integrated life events calendar component. Fieldwork started with web mode, followed by "early CATI", i.e. prioritised cases of respondents who were least likely to complete online due to a lack of internet connection or they never use the internet, or they do not have an email or mobile number.

This is our first Blaise 5 study with an integrated calendar component, which is used to collect detailed information about important events that occurred earlier in the respondent's life.

Issues with Calendar in CAWI mode, due to Load Balancer

After several iterations of development, it was tested in our Staging environment and signed off as working. However, once it was implemented in Production, there were issues with some respondents getting errors as they interacted with the calendar.

After much investigation it was determined that the load balancer in Production was interfering with the operation of the Blaise survey and the corresponding calendar. There were also issues with the Session database relating to this and timing out in some instances.

The main issue seemed to be an interaction of the calendar application and the session data. The session data is temporary or secondary storage of the case when the case is being entered. The session data is read by the calendar application so information can be displayed in the calendar portion of the screen. On the Production server during an interviewing session the connection to the session data was lost. The Blaise Team thought that this was an issue with the load balancer. The production

server park has 3 web servers so when a respondent connects to the project, they are assigned to one of the web servers to spread the load.

To solve the issue, we by-passed the load balancer by re-directing respondents to a named server. We did this by setting up a re-direction on the server and by updating the URLs in the respondents' direct email links.

The Staging environment was configured a little differently than the Production environment, with only 1 web server instead of multiple as in Production and there was no load balancer. So, the issues were not encountered in Staging as they were in Production.

From the time the issues were identified, we were forced to take down the survey, to investigate what was happening, and resolve the issues. A temporary work-around was developed, targeting each respondent to a particular web server to run the questionnaire and the integrated calendar, thus bypassing the load balancer. Only then were we able to relaunch the survey, send new communications to respondents, asking them to participate and providing new links to the survey.

Statistics Netherlands / Team Blaise helped develop this function and the integration of it with our Blaise 5 survey. They also assisted in troubleshooting and resolution when we had issues in Production.

Performance Issues, Lag Between Screens

Once the CATI fieldwork started, we experienced performance issues. Telephone Interviewers reported issues about the lag they were experiencing when using the Calendar.

After some investigation it was determined that the Launcher's project setting needed to be changed and use the CSS Grid Web Rendering Option. This setting was already present in the Topic instrument and in CAWI there was no lagging when the calendar was being used.

Without this setting in the Launcher project, the performance of ELSA Life History (with Calendar) suffered in CATI mode. We discovered that when you have a "chained project" such as a Launcher and Topic, there is a lot of **inheritance** so the settings in the initial project, (i.e. the Launcher) overrides settings in the Topic.

4 The Future

4.1 Single Database for All Modes

The goal for multi-mode surveys is to move to single instrument and database approach for all modes, but there are challenges that must be solved before then. In particular, it is necessary to define a comprehensive set of rules to reconcile updates from a variety of sources and across various modes before applying them to the master or historical database. Then a solution can be developed, tested and implemented.

Among the considerations there must be rules for dealing with partially completed web or CAPI interviews, opt-out or ineligible cases initiated by respondents contacting the Survey Enquiry Team (i.e. Help Line), and cases deemed ineligible as a result of initial questions in the survey itself. As stated previously, Web and CAPI are typically open at the same time and CAPI interviews are done offline and later uploaded to the server. Moreover, data will need to be merged not only from different modes, but from different sources (Blaise main, session or launcher databases, the active and historical databases on the server and from the Survey Control System (SCS)).

There needs to be a process to copy both Web and CAPI data to an intermediate backup or merge

history database and from there select the appropriate cases to keep current, according to a comprehensive set of rules.

At the University of Michigan, they have implemented Write Interceptors, and their rules are very complex.

4.2 Multi-Person, Multi-Mode Household Surveys

The National Diet and Nutrition Survey (NDNS) is an example of a multi-person, multi-mode household survey that we are conducting.

The NDNS is a cross-sectional, continuous survey of diet and nutrition for individuals, designed to be representative of the general UK population, and aiming to provide government and other departments with the information they need to develop policies and monitor public health and food safety.

In terms of eligibility to take part in the survey, the sample is split into 'main sample' and 'young persons sample'. In the main sample, up to 2 adults aged 19 and over and 1 young person aged 1-18 are eligible to take part, and in the young persons sample, up to 2 children (self-selected if there are more) aged 1-18 can take part, adults are not required in this sample type.

The survey is CAWI with a CATI opt-in, where telephone assistance can be requested by respondents. Each person in the household has their own record to complete the survey and corresponding dietary recalls. However, the survey instrument does look across all person records of a household to check the status of cases.

Generally speaking, when conducting CATI or CAPI interviews, it is desirable and more efficient to access all persons in a household to interview them at once, but for web mode it is desirable and better for data security and privacy for each person to access their own record.

We are struggling to determine the best approach to achieve these goals.

5 Lessons Learned

We have learned a great deal through the experience of doing these varied surveys in Blaise 5. Here is a summary of some of the key learnings.

Mixed-Mode Survey using Separate Instruments and Databases

There is a need to keep the CAPI survey updated with results from the Web survey, before and during CAPI fieldwork. During CAPI fieldwork, once interviewers downloaded cases allocated to them, there was a need to send email and text notifications of results in the web survey to avoid interviewers visiting respondents that already completed online or opted out of the survey.

It is also important to manage and reconcile updates from the Survey Enquiry Team (i.e. Help Line) vs direct updates from web respondents or interviewers completing the questionnaire.

There may be a need to reissue access codes if a respondent lost them or didn't get the letters but want to complete the survey online on their own. This is to avoid data security and privacy issues. It may be possible to plan for this in advance, as was done on the WNTS survey, and create spare records and access codes that could be given out in these circumstances.

Load Balancer Interfering with External Components

We experienced issues with the interactive events history calendar in the ELSA LH study. Related to this we also had issues with the Session database timing out.

The Staging environment, where testing is performed, must be configured like-for-like to the Production environment, in terms of the number of Web servers, load balancer and other configuration. Currently on the Production cluster we have 3 web servers with a load balancer and one Management server, but in Staging we have only 1 web server without a load balancer.

Also, it was suggested by the Blaise Team to split roles over multiple servers when a role uses a lot of resources (e.g. DayBatch creation).

Calling External Files within Blaise CAPI surveys

The location of external files within Blaise CAPI program must be kept with the main program and not in a folder (such as C:\WinCMS\Incentive), to ensure they can run on all devices and platforms). This is different than what is needed for Blaise 4.

AppSupport Settings

We have become more aware that when you create a new package (.bpkg) of a survey with a project that has 'app support' set to "on" in the Server Manager, the App packages (*.app.bpkg) should also be created on the server (in c:\blaise5\surveys\SurveyRoot).

Blaise Version Upgrades

Like any new software you might encounter issues you don't anticipate, and when bugs are found, you might need to upgrade a version of Blaise. We were not able to upgrade the server without a lot of effort (live projects were running and we needed to minimise interruptions in the interviewing work).

It is difficult to have a "stable" version, as it requires balancing the desire for enabling new features against staying with a stable version.

Blaise Support were quick in helping us resolving issues. Some were genuine bugs with the software.

Inheritance

When you have a chained project such as a Launcher and Topic, there is a lot of inheritance so the settings in the initial project (e.g. the Launcher) overrides settings in the next project (e.g. the Topic). These things must be checked carefully.

In the old implementation of ASP.NET the inheritance behaviour was different. In MVC, the root where you start (in this case the login instrument) is leading for certain functionality, e.g. when you switch to another survey/root (with the StartSurvey action).

This is the same for the custom MVC application (included in the ELSA .bcps file), it "looks" in the folders of the questionnaire you started first (if it's the login, for example, in c:\blaise5\surveys\LoginSurveyRoot).

For ELSA and its Life History Calendar, we had to install the calendar components in all potential entry-points (the main instrument, the login and the launcher instruments) to make the application load since all projects inherit some shared resources/styles and other functionalities.

CSS Grid Settings

In the ELSA LH project, we experienced performance issues on the CATI survey and found that the CSS Grid Web Rendering Option needed to be modified. This setting was already present in the Topic instrument and in CAWI there was no lagging when the calendar was being used.

Without this setting in the Launcher project, the performance of ELSA Life History (with Calendar) suffered in CATI mode. We discovered that when you have a “chained project” such as a Launcher and Topic, there is a lot of inheritance so the settings in the initial project, (i.e. the Launcher) overrides settings in the Topic.

Session Database

We have a better understanding of Session Data and are aware that we need to find a solution to the problem of the Survey Control System API updates being overwritten by the Session Data.

Hard Appointments Not Delivered due to Maximum Calls

During the ELSA Life History we noticed that some hard appointments did not get delivered. This occurred when cases reached the maximum number of calls (these cases had “Certain date” appointments in the “Case Info” tab on the Dashboard and the “Filter Result” column read “Number of calls”).

The Blaise 5 Help on the CATI Specification page states that the Maximum number of calls “(...) *setting is ignored for numbers with a hard appointment or with a preference appointment that is current.*”. We notified the Blaise Team about this there should be a change in future versions of Blaise.

DayBatch Creation Performance

We are aware that the DayBatch creation uses a lot of resources and that we should split roles to ensure the available memory on the server isn’t adversely affected by these processes.

Using the DayBatch select filter to exclude cases when manually getting cases suffered a performance penalty. Therefore, with the version of Blaise used in Production during fieldwork (5.13), creating the DayBatch had an impact on the available memory of the server. The latest version of Blaise comes with performance improvements in this area.