Implementing an Office & Field Survey Management System for PIAAC using Blaise

Jacqueline Hunt, Central Statistics Office, Ireland

Introduction

This paper describes the Blaise developments undertaken to facilitate the Central Statistics Office (CSO) participation in the Programme for the International Assessment of Adult Competencies (PIAAC). PIAAC is a collaborative project between governments and an international consortium to assess cognitive skills, IT literacy and formal educational attainment. The study is taking place across twenty-five OECD and partner countries in 2011-2012 with results due to be available in 2013. The PIAAC consortium's assessment instrument was designed to maximise cross-cultural, cross-national and cross-language validity. The application was developed as a "black-box" solution designed to run in its own virtual machine. All participating countries had to adhere to common technical standards when implementing the survey but each country was responsible for its own survey and case management functionality. The final CSO solution was developed almost exclusively in Blaise using a combination of Maniplus and Manipula scripts. This paper describes the design approach, the functionality provided in the various GUIs and the data management features.

1 PIAAC Applications

The PIAAC applications supplied by the consortium consist of a PIAAC Virtual Machine (VM) and the TAO platform that runs inside the virtual machine.

The TAO platform contains all the necessary software to run the following three stages of the PIAAC assessment.

1. The global PIAAC interview workflow (WF) describing the case initialisation, the disposition codes, the ICT-Screener, ICT-CORE and ICT-Tutorial; the navigation among instruments; the booklet selection controls, and the interviewer instructions for Paper-and-Pencil booklets;
2. The background Questionnaire (BQ); and
3. The Cognitive Instruments (CI) including general and domain-specific orientations.

The TAO platform collects and manages all the PIAAC data. Statistical offices taking part in the study were instructed that no additional software should be installed inside the PIAAC official VM. External software could be installed on the host machine, outside the virtual machine, to manage the survey processes.

The PIAAC data is stored as case objects; the case objects can be imported to or exported from the TAO platform in a zipped XML file format. If an invalid XML file is provided to the system, it will not be imported and the case will not be available for interview in the TAO. On completion of a PIAAC interview an object based [PERSID].zip archive is produced and exported to an output folder on the host machine.

A series of scripts were provided by the consortium to access the data from the TAO platform. The scripts can be triggered from external applications using the Case IDs as parameters to select specific cases. Table 1 is a list of some of the scripts provided.

Script	Function
Basic scripts	
StartCAPI(optional:'new' or PERSID)	Starts the VM and the TAO application
ResumeCAPI(PERSID)	Starts the VM and the TAO application, resumes the interview specified by the PERSID.
ExportResult(optional:PERSID)	Copies all available data from the PIAAC VM to the Windows environment. If the result data of a certain interview is required the case can be specified by adding the PERSID.
StopVM	Terminates the currently running VM.
HandleCAPI	Starts or resumes the current virtual machine with the given PERSID, dependent on the state of the interview inside the VM.
Advanced scripts	
DropCase(PERSID)	Deletes the interview specified by the PERSID from the TAO database.
DumpCase(PERSID)	Activates to dump an interview specified by the PERSID within the VM and copies the resulting SQL file to the Windows environment. The interview is exported in its current state. The output can be used to Restore a case.
ImportCase(PERSID)	Copies a dump file generated from the DumpCase script from the Windows environment to the VM and imports it to the TAO.
DumpAllCases	Activates to dump the entire TAO database and copies the resulting SQL file to the Windows environment.
RestoreAllCases	Copies a dump file of the entire TAO database to the VM.
GetCaseState (PERSID)	Gets the states of a certain case or of all cases from the VM to Windows. If getting the state of a certain PERSID the return value is set to the cases running state.

Table 1: VM Scripts

2 Lessons from the Pilot

The office had a very tight timetable to prepare for the pilot study scheduled for April 2010 due to late official confirmation to participate in the survey. This resulted in a very short development time of five months, starting at the end of October 2009. The limited time available meant we had to prioritise what could be delivered. A lot of time was required in the beginning to become familiar with the installation and administration of the consortium software. Due to the limited time available for any new development it was decided that the pilot would be conducted with the existing survey management applications that were used for the Quarterly National Household Survey (QNHS) and the Survey of Income and Living Conditions (SILC) see Fig. 1, for the list of software tools used in these systems. The intention here was to reduce the development work required by reusing existing software and prioritise any specific requirements necessary to conduct the survey.

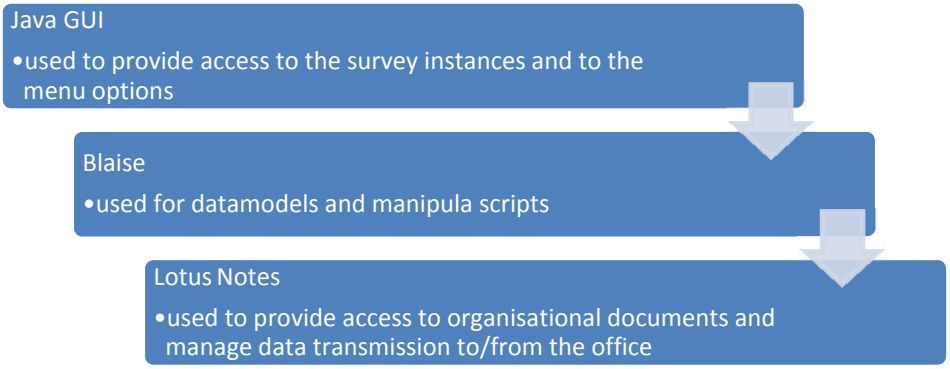

Figure 1: Existing HSCU Survey Management Application Components

Signification changes to the survey management applications were required to adapt them to work with the PIAAC VM and its object based file structure. The PIAAC case files are stored as objects where as the QNHS and SILC survey data are stored as Blaise databases.
The other essential requirements identified for the pilot included the following;
- a preliminary questionnaire to identify the Household composition,
- a selection function to identify the appropriate survey respondent from the Household composition,
- data from this questionnaire to be accessible to the PIAAC assessment,
- an administration block to record progress, and
- report functionality based on the PIAAC disposition codes.

The pilot study was considered a success by the survey area based on the number of completed assessments successfully collected. However, there were a number of problems reported with the field applications that were attributable to the lack of integration between the PIAAC system and the survey management applications. Some of the issues were known in advance of the pilot but there was no time available to correct them. One of the difficulties was the poor transition between completing the Screener and starting the PIAAC assessment. The Screener is the Blaise datamodel that was used to collect the household details and select a random person from the eligible residents to complete the PIAAC assessment. The interviewers had to remember to prepare an input XML file before starting the assessment to ensure individual details would be imported into the PIAAC TAO. The main problems reported by the field interviewers after the pilot were usability and functional issues. There were some issues with performance and screens freezing during the PIAAC assessments, these were resolved by the consortium in advance of the live survey. There was also some confusion and difficulties accessing and closing the VM, which sometimes resulted in multiple VMs being open at the same time. Another issue was that the Screener household data had not been tied down sufficiently and could be changed after the random selection had taken place. This resulted in a number of problems with mismatched data. The link between the PIAAC assessment and the survey management applications was limited which meant that the interviewers could not view the status of their PIAAC cases from the survey management application. Addressing these issues for the live survey required a complete re-design of the survey management applications. The key lesson learned from the pilot was that the field applications required much tighter coupling with the PIAAC "black-box".

3 Live Applications

3.1 Requirements

The survey area IT requirements for the live survey were similar to the pilot survey, they included

- A facility to capture the household constituents and randomly select one eligible person to complete the PIAAC assessment
- Para data capture to include the number of contacts with the household and the outcome of each contact
- Single point of access for all data entry components
- Laptop GUI for interviewers to manage their workload, display case details, and transfer updated cases to the office
- Data management and administration including management of data transfer from the field to the office
- Field supervisor and office GUI and reports.

Based on the pilot feedback and the difficulties reported, the key features of the new design had to encompass

- Better integration between the CSO data collection, the survey management applications and the PIAAC applications
- Automation of as many user tasks as possible including starting and stopping the PIAAC VM
- Enhanced access and views of the data including an up to date case status for the entire case.

3.2 Sample

How to distribute the sample among the field staff was one of the first issues to address. As this was a fixed survey the sample households were known in advance. A sample file was produced with the Interviewer IDs assigned to each case based on their geographic location. The same sample lookup file was installed on all the interviewer laptops. The interviewer menu was used to filter the sample file to display only relevant cases for an individual interviewer. The interviewers could also filter their own work allocations to create different views of the cases, for example, views based on the case sequence number or the case status. Functionality to amend the sample lookup file and prepare it for redistribution to the field if the work allocation needed to be altered was added to the office menu.

3.3 Data Collection

The data collection had a three phase approach. During the initial contact with the household the interviewer attempted to interview at least one of the residents to identify all the individuals living in that household. When the personal details were collected the PIAAC respondent was randomly picked from the number of eligible people in the household using the Blaise random function. Eligibility was based on the criteria that the individual was between the age of 16 and 65. A Blaise datamodel called the Screener was used to collect the household details, confirm the age criteria and select the person to complete the assessment based on the Blaise random function.
The next stage in the data collection process was to complete the PIAAC assessment. For the live survey we needed to restrict the interviewer's access to the PIAAC VM. During the pilot the interviewers had to select between conducting a *New Assessment* and *Resuming* an existing one. Resuming an interview brought the interviewers to a "Welcome Screen", see Fig. 2, where they could select the case to resume. This screen also contained a button to start a *New Interview* that could not be disabled.

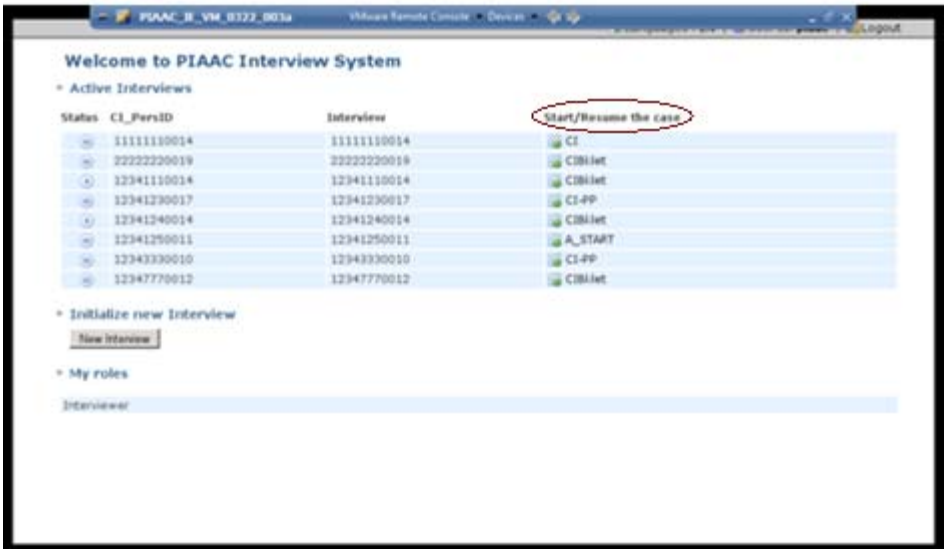

Figure 2: PIAAC Welcome Screen

Allowing the interviewers direct access to the VM resulted in a number of incidents of mis-matched Case IDs as the interviewers could key new Case IDs directly into the TAO via the *New Interview* button. We needed to find a better solution to access the PIAAC assessment regardless of it being a *New* or *Resuming* case. As we did not know in advance who the household constituents were and who would be selected to complete the assessment it was not possible to pre-fill the PIAAC VM with any individuals' details. This meant that the input XML file for all new cases had to be prepared after the Screener details were complete. A Blaise Alien Procedure was used in the Screener datamodel to create the XML file that could be used to pass data directly into the PIAAC VM without any interviewer intervention. The Screener writes the information relating to the selected individual such as age, gender, highest educational qualification to a new input XML file, using the PERSID field as the file name. The file name is then used as a parameter to launch a case in the PIAAC VM with the imported case details.

PIAAC2011Scr.bla | Review.inc | Procedures.inc | CreatePIAACInputZipFile.man

```
Description    : CreatePiaacZip - Used to create a zip file which will be used by the Piaac VM Questionnaire

*****************************************************************************************************************************************

PROCEDURE CreatePiaacZip
PARAMETERS
   TRANSIT BlockNo, LDU, PiaacID, Name, Age, Phone, Addr , BookID_PPC, BookID_PP1, BookID_PP2, BookID_PRC : STRING
   TRANSIT Sex, SexCode : TSex
ALIEN ('CreatePiaacInputZipFile.Msu', 'MakeZip'){ Manipula call }
ENDPROCEDURE
```

Figure 3: Procedure Call Code

A new AutoIt (Automation and Scripting Language) script, called LaunchPIAAC was developed using the consortium scripts to handle the process of opening the VM, starting a new or resuming an existing case, writing the status result to a file on the host machine, exporting the case to the host machine and finally, closing the VM. This restricted and controlled the interviewer's access to the VM and prevented any case being created without a corresponding input XML file.

The consortium scripts used in LaunchPIAAC are:

- HandleCAPI
  Starts the VM and relevant case at the last point of the assessment. It calls the StartCAPI/ResumeCAPI/GetCaseState/ImportCase scripts with the PersID passed as a parameter from the Blaise screener.
- GetCaseState
  This script continuously polls the state of the case in the VM while it is open, the last state is exported to the file c:\piaac\piaacstatus.txt file.

- DumpCase
  If the GetCaseState returns a status of *'paused'* or *'finished'* this script dumps the case and copies the resulting SQL file to c:\piaac\administration.
- ExportResult
  If GetCaseState returns a status of *'paused'* or *'finished'* this script exports all case data and copies it to c:\piaac\output\PERSID.zip as a completed case.
- StopVM
  Shuts down the VM.

The PIAAC assessment started with some initial CAPI questions asked by the interviewer followed by the main assessment collected via CASI (Computer Assisted Self-Interviewing). Fig. 4 is a screen shot from the early part of the PIAAC interview.

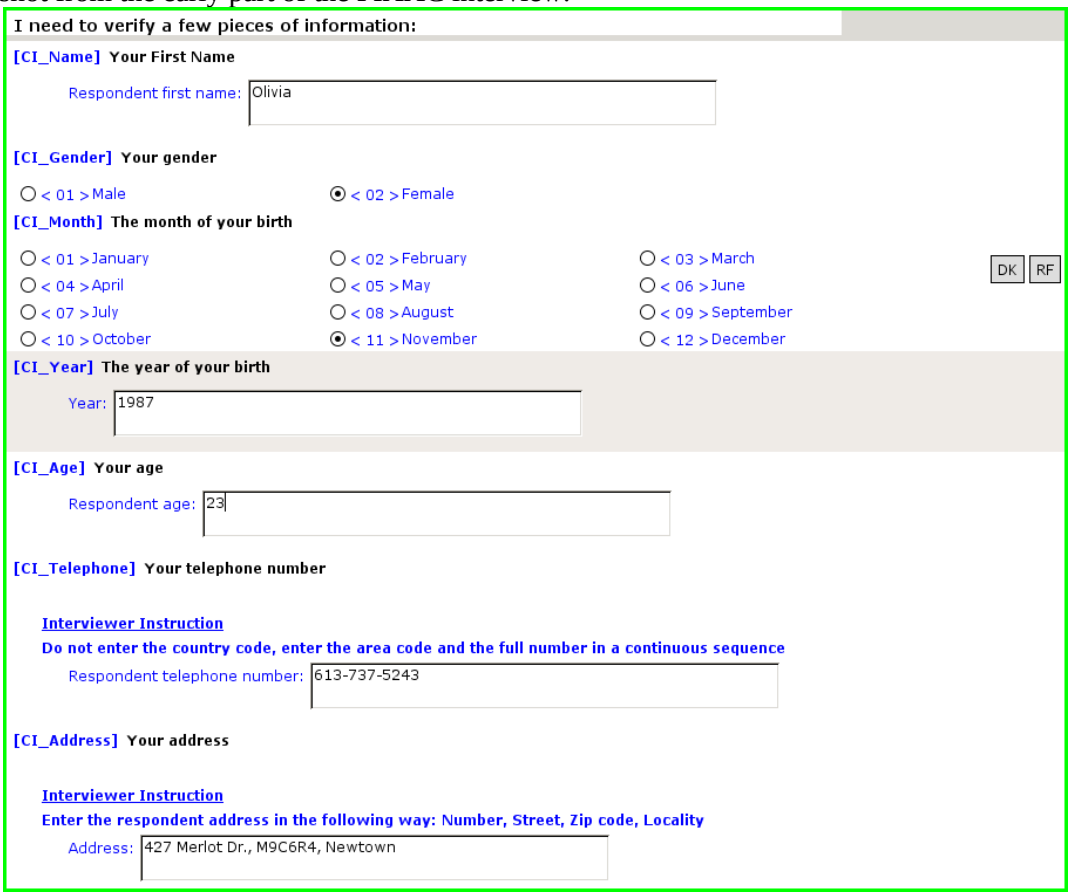

Figure 4: PIAAC Srceen

The third and final part of the data collection was the para data section. Another Blaise datamodel, called the Admin was used to collect the survey case's call histories. Fig. 5 is a screenshot of the Call History section of the datamodel. During the pilot only one datamodel had been used to collect the household components and para data information. For the live survey two datamodels were used to provide more flexible access to the datamodels particularly the Admin as this can be accessed multiple times by the interviewers to record contact and case status information.

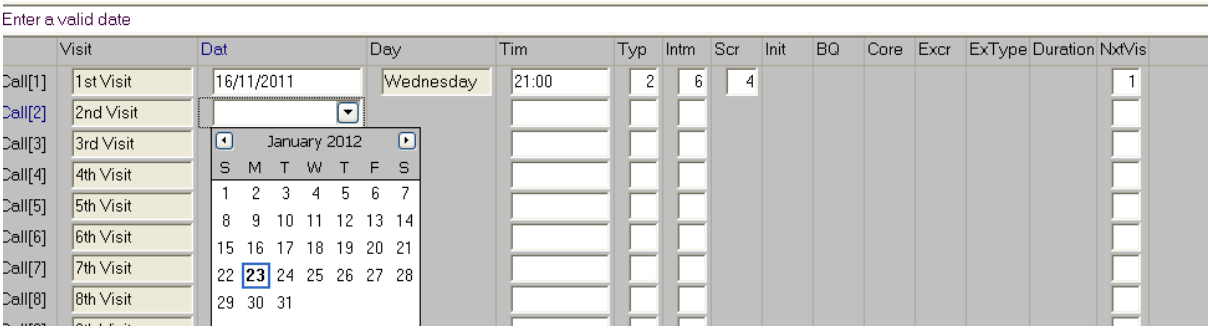

Figure 5: Visit Data Entry

The interviewer data collection workflow is described in Fig. 6.

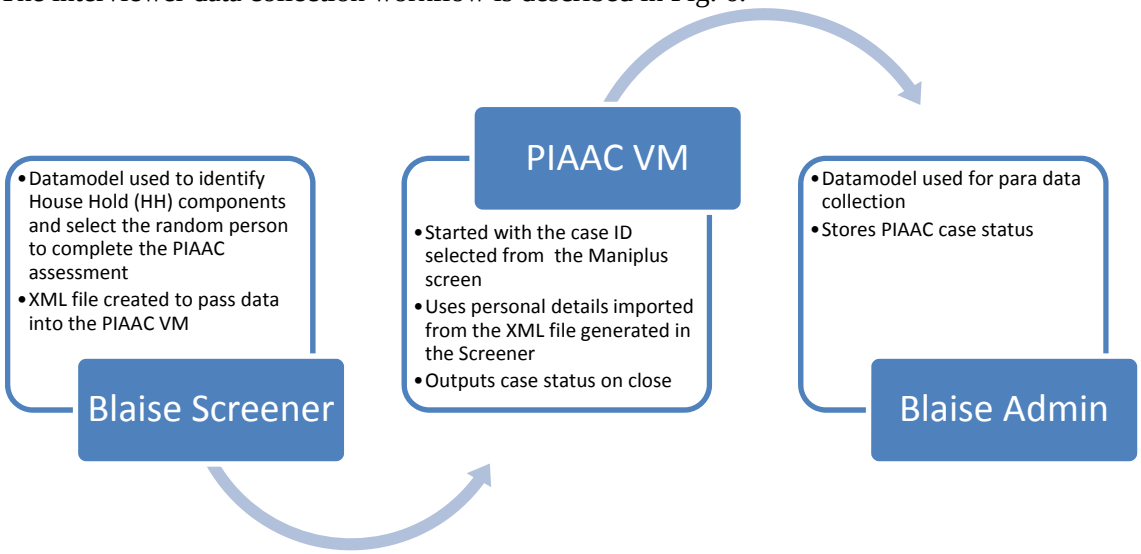

Figure 6: Data Collection Workflow

The interviewer's data collection workflow is described below:

(i) Complete Screener.
(ii) Screener application selects eligible respondent for interview.
(iii) If selected person is available, proceed with PIAAC Background Questionnaire, Exercise and Observation Module (ZZs).
(iv) Shut down laptop.
(v) Complete Gratuity Card detail and leave household.
(vi) Update Admin as soon as possible.
(vii) Transmit Data to CSO
(viii) If Selected respondent is unavailable, ensure contact details are correct.
(ix) Contact person and arrange an appointment as soon as possible.
(x) Record contact information in the Admin.

The interviewer could access the Screener, Admin and PIAAC components from a TempTable in the Maniplus application, see Fig. 7. Selecting a row in the table followed by the component will identify the Case ID and open the relevant case in either of the Blaise datamodels or the PIAAC VM. The information displayed in Fig. 7 is taken from three sources; the Blaise Screener datamodel, the Admin datamodel and the sample lookup file.

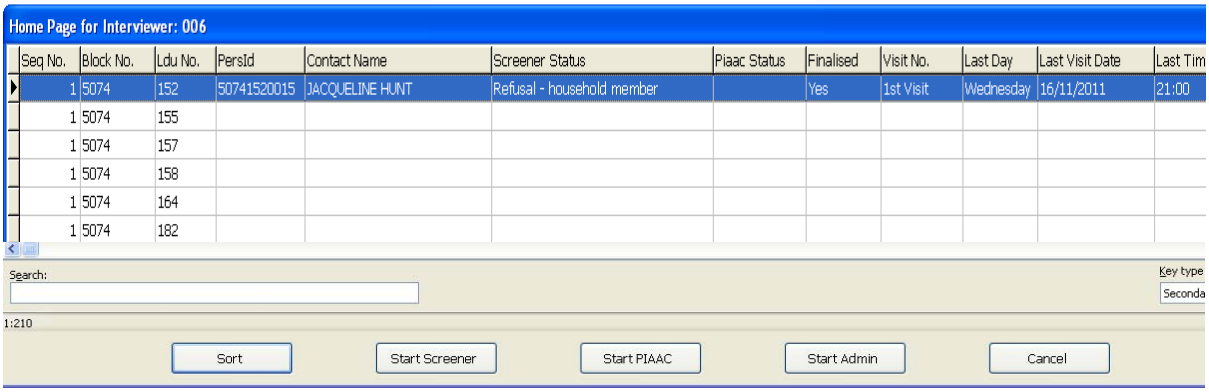

Figure 7: Survey Home Page displaying the list of cases

3.4 Survey Management GUIs

Blaise Maniplus applications were used for all the interviewers and management menus.

3.4.1 Interviewer Application

Interviewers could view their case load, access the survey instruments, view appointments, manage their cases to transfer to the office and view transmission logs and progress reports. Starting any of the data entry components from the Home Page was a straight forward procedure for the interviewer. The interviewer selected the case from the TempTable, see Fig. 7 and the case ID was passed into the datamodel or the PIAAC VM as a parameter.

Traditionally, when transmitting data to the office we have used the Blaise form status of *New* and *Changed* to select the cases for transfer. This worked for the Blaise Screener and Admin data as we retained the Blaise database structure. However, a way of replicating finding *New* and *Changed* PIAAC cases was required. This was resolved by initiating an action in the Maniplus script to copy the PIAAC output zip file to a staging folder on the local machine as the PIAAC VM was being shut down. When an interviewer initiated a data transfer to the office the application would pick up any *New* or *Changed* Blaise records and any PIAAC case files that were held in the staging folder. The staging folder is cleared after every transmission so only cases that had been accessed in the PIAAC VM since the last transmission were transferred to the office. A Transaction Log was created after every upload of data for the interviewers to view the Case IDs and component i.e. Screener, Admin, PIAAC that were successfully transmitted to the office. In addition to the *New/Changed* data transfer the interviewers could also select specific cases to transfer, see Fig. 8. This proved to be a useful option for the field and office staff.

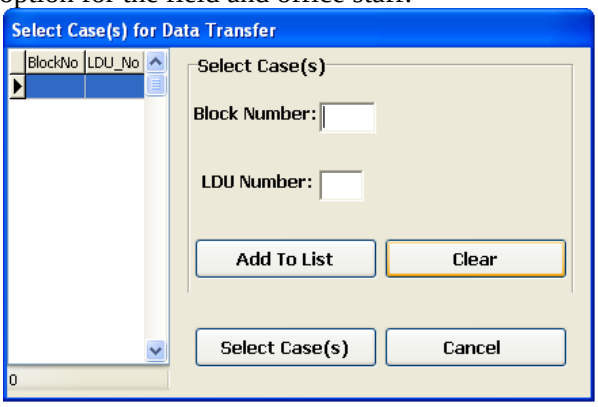

Figure 8: Select Cases to Transfer to the Office

To provide a case status field for the interviewers and management reports we needed to have access to the PIAAC case status. A PiaacStatus.txt file was created when a case was exited that contained the latest status for the last open case. This file had a generic file name and contained only the last active case status. We wanted to find a way to save this information with the rest of the case details. We decided to save this information to the Admin database. If the Case ID already existed in the datamodel the status field was updated. In some circumstances the Admin database would not have a corresponding case, this occurred when the PIAAC selected person was available for interview during the initial contact with the household. In this instance the interviewer would not have had an opportunity to add any details of the case to the Admin database. In this scenario the Case ID i.e. the Block and LDU along with the PIAAC status were written to the Admin database.

3.4.2 Management Applications

The supervisor functionality included the facilities to view their teams' progress and interrogate specific cases. The latest information that had been transmitted to the office was available to the supervisors each morning. Fig. 9 displays the summary information available to the supervisors which is similar to the status information displayed to the interviewers.

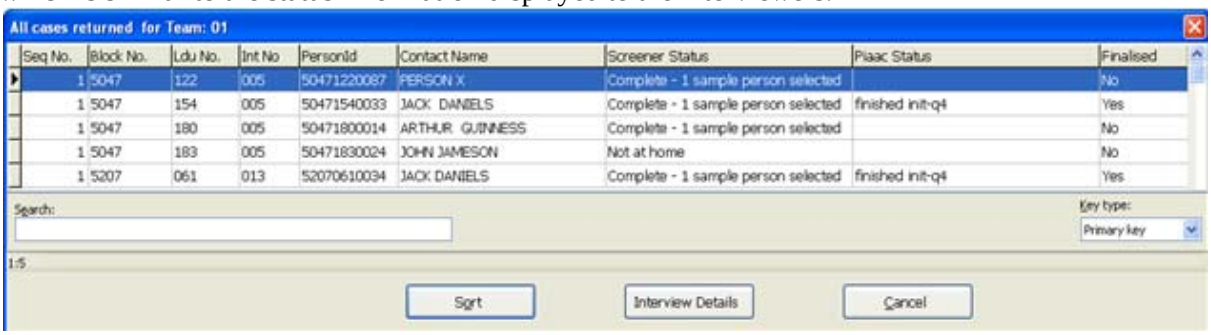

Figure 9: Co-ordinator View of Cases

The supervisors could view a more detailed summary by selecting the case. Different views of the cases were also available and they could be filtered by variables such as the completion status, *Finalised* and *Incomplete*, and Interviewer ID.

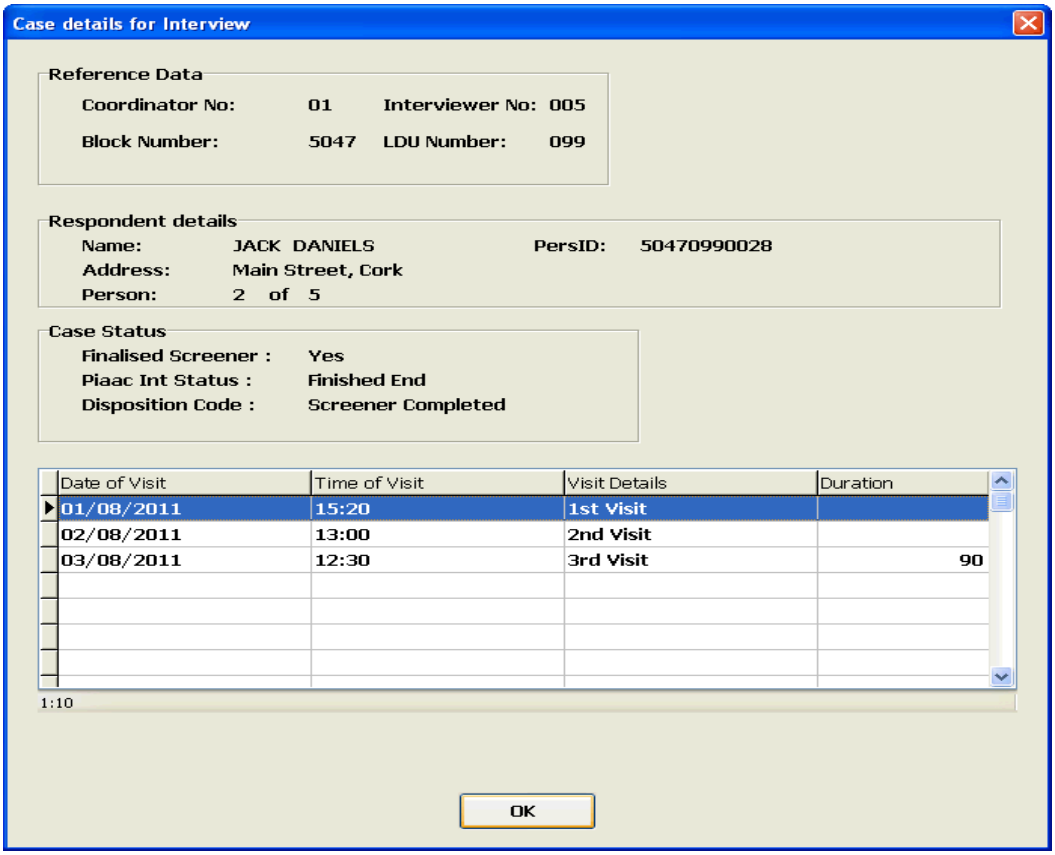

Figure 10: Case Details

The supervisors could also view and print various reports on the data that were accessed via MS Excel. Outputting the reports in MS Excel allowed the users to filter and sort the data as required. The application used by office based staff had similar views of the data and reports. In addition, they could update the sample file and prepare it for re-distribution to the field staff.
Any functionality not available directly in Blaise was accessed from a utility created by our Java team. This utility covers functionality such as encrypting, decrypting, zipping, unzipping and general file management such as copying, moving and deleting files.

3.5 Append Job

A suite of Maniplus and Manipula scripts were used to manage the data when it was received in the office. Each interviewer's lodgement was stored as a folder on the network. A control file with a list of the folder names was used to guide the Maniplus application. The Maniplus script worked through each interviewer's lodgement, first unzipping and decrypting the files. Any Blaise data was appended to the office Blaise Screener and/or Admin databases and the PIAAC zip files were moved to a network folder ready to be imported into the PIAAC Data Management Expert (DME), the data processing application provided by the consortium. Transactions logs were created showing all the records added for each interviewer. The job also prepared four update files for the supervisors based on their own teams' lodgements that were available for them to download each morning.

4 Conclusion

The PIAAC survey has been running since August 2011 and is due to complete in March 2012. Part of the fieldwork finished in February 2012 and some review sessions have already taken place. The initial feedback on the applications is a lot different to the pilot experience. Most of the feedback was very positive with the interviewers reporting high satisfaction levels with the applications. The facility to select cases for transfer was highlighted as a useful feature. The only suggested improvements were to add a Comment field to the Admin datamodel and to revise the wording of the outcome codes, both very minor suggestions. Reviewing the issues logged to the Helpdesk over the data collection period shows most of the incidents reported were due to poor connectivity which is outside the scope of these applications.

Before these developments we had limited experience using Maniplus. Our experience with the software has been very positive and we were very pleased with the functionality we were able to deliver. We are hoping to reuse the applications with minimum amendments for an upcoming new household survey.

5. References

Kuusela, V. & CMS BCLUB Working Group (2010) , Features of Case Management in CAI Systems

Tudor, H (2009), PIAAC – TAO Data Import and Export, General Technical Processes and File Structures, Centre de Recherche Public

6. Appendix

LaunchPiaac Script

```
; ###############################################################################
; Name: LaunchPIAAC
; ###############################################################################
; History
; ###############################################################################
; 12.04.11 : Created
; ###############################################################################
; Description
; ###############################################################################
; This script is executed when the 'Start PIAAC' button is selected for the
; relevant Person ID in the Home Page of the Blaise 'Programme for International
; Assessment of Adult Competencies'. It starts or resumes the current PIAAC
; virtual machine (VM) with the given person id dependent on the state of the
; interview inside the VM. When the interview is paused or finished the case
; status is written to c:\piaac\piaacstatus.txt file. Case data is dumped to
; c:\piaac\administration\'persid'-dump.sql and also exported to
; c:\piaac\output\'persid'.zip. Then the VM is shut down and control is
; returned to Blaise.
; ###############################################################################
; PIAAC Consortium scripts used in this program:
; ###############################################################################
; HandleCAPI
; GetCaseState
; DumpCase
; ExportResult
; StopVM
; ###############################################################################
; Parameter: Person ID
; ###############################################################################
; ###############################################################################
; Executing HandleCAPI - starts VM and relevant case at relevant point of interview
; HandleCAPI calls StartCAPI/ResumeCAPI/GetCaseState
; Person ID passed as parameter from Blaise
$persid = $CmdLine[1]
$Program = "c:\piaac\handlecapi.exe "
$Arguments = ' ' & $persid &''
RunWait($Program & $Arguments,"",@SW_HIDE)
; ###############################################################################
; bringing Virtual Machine to the front
if(WinExists("[Class:VMPlayerFrame]")) Then
    WinActivate("[Class:VMPlayerFrame]")
endif
; ###############################################################################
; Executing GetCaseState - to frequently check state of relevant Case in the VM
; Person ID passed as parameter from Blaise
; PersID, current activity and state of case in the VM are copied to the clipboard
; NB in c:\piaac\piaacscriptconfig.ini Behaviour must be set to Frequently Update
$Program = "c:\piaac\GetCaseState.exe "
$Arguments = ' ' & $persid &''
Run($Program & $Arguments,"",@SW_HIDE)
sleep(30000);30 second delay to allow update of checked status
; ###############################################################################
; Getting relevant state data from clipboard
$bak = ClipGet()
$result = StringRight($bak, 9)
$Status = StringStripWS($result, 8)
; ###############################################################################
; Checking for state of case in the VM, if case 'paused' or 'finished' then
; dump, export case and stop the VM. If case is 'running' keep checking state.
While $Status = "Running" ;use infinite loop since ExitLoop will get called
  $bak = ClipGet()
    $result = StringRight($bak, 8)
    $CheckStatus = StringStripWS($result, 8)
  If $CheckStatus = "Paused" Then ExitLoop
  If $CheckStatus = "nished" Then ExitLoop
WEnd
; ###############################################################################
; Writing Person ID, current activity and case state from clipboard to text file
```

```
$file = FileOpen("c:\piaac\piaacstatus.txt", 1)
; Check if file opened for writing OK
If $file = -1 Then
   MsgBox(0, "Error", "Unable to open file.")
   Exit
EndIf
#include <File.au3>
_FileWriteToLine("c:\piaac\piaacstatus.txt", 1, $bak, 1)
FileClose($file)
; ###############################################################################
; Stopping execution of GetCaseState
ProcessClose("getcasestate.exe")
; ###############################################################################
; Executing DumpCase - dumps a case of the database in the current VM and copies
; it to the host machine
; Person ID passed as parameter from Blaise
$Program = "c:\piaac\dumpcase.exe "
$persid = $CmdLine[1]
$Arguments = ' ' & $persid &''
RunWait($Program & $Arguments)
; ###############################################################################
; Executing ExportResult - all case data is copied to the host machine for
; further processing
; Person ID passed as parameter from Blaise
$Program = "c:\piaac\ExportResult.exe "
$persid = $CmdLine[1]
$Arguments = ' ' & $persid &''
RunWait($Program & $Arguments)
; ###############################################################################
; Executing StopVM - powers down the PIAAC VM and stops VMWare Player
RunWait('C:\piaac\stopvm.exe')
; ###############################################################################
; End LaunchPIAAC
; ###############################################################################
```