# Section E. Organisation

## Using Blaise in a survey organisation where the researchers write the Blaise datamodels

*Tony Manners, Office for National Statistics (UK)*

CAI, starting with CATI, has been with us well over two decades. Most of the original questions it posed for survey organisations have been resolved[iii]. The kinds of organisations and the types of work for which it is most likely to be successful are well known[iv]. While most organisations have a range of CAI modes in their portfolios, to use as appropriate, some broad generalisations can be made. CATI and CASI, with rapid design, programming, implementation and reporting of simple instruments, are well established in the commercial field, particularly in the USA[v]. CAPI has so far seemed best suited for organisations which typically implement lengthy, complex instruments on social and economic topics for government and other parts of the public sector, and do so on a sufficient scale to justify a permanent staff of interviewers. Such justification is most likely if their portfolios include a number of regular or continuous surveys. This paper concerns one of the few remaining questions about CAI on which different survey organisations, including those which share many characteristics, retain different and strongly held opinions: what kinds of staff should author CAI instruments.

Under paper and pencil interviewing (PAPI), questionnaires were usually designed by the researchers who would eventually analyse the results themselves or who liaised closely with the customers' analysts to design questions and instruments to meet their analysis needs. Programming the routing of a paper questionnaire could be conceptually sophisticated, but its physical process was often a simple matter of using scissors and paste.

PAPI questionnaires included only very rudimentary interviewer's instructions and edit checks, if any at all, but we should note how these survey elements were dealt with since they were to become integral parts of the CAI instrument. Detailed interviewers' instructions were designed by some combination of researchers and fieldwork specialists. Interviewers were asked to study them ahead of the fieldwork and remember them in appropriate circumstances. Edit checks were, of course, carried out after the fieldwork was finished. They concentrated mainly on routing and ensuring that answers were within valid ranges, since errors of these types were possible at every question in the instrument. Data capture and editing was often carried out in third generation languages on mainframe computers. For all these reasons, edit checks tended to be a matter for computing specialists, with great variation between researchers in the degree to which they were involved. Looking at PAPI as a whole, including the elements which were already computerised before CAI, like editing and analysis, we can see a range of approaches to the division of labour between researchers and computing specialists. These approaches varied between organisations at any one time, and also within the same organisation over time. At one extreme, researchers simply specified their editing and analysis requirements to computer programmers. At the other extreme, researchers sought to extend their control of the survey process by specifying directly in higher-level languages like SPSS or SAS. Over time, as PCs spread in the workplace and PC software became increasingly easy to use, researchers have become accustomed to hands-on control in areas of the survey process which they may not have had in former years. For such researchers, CAI appeared a natural progression in their work with computers.

Computerisation of the questionnaire, and extension of the questionnaire to cover the functions previously carried out in post-fieldwork edit programs, posed a strategic issue for organisations (at least implicitly). Was this innovation to be regarded as essentially a development in the field of computing or as essentially a development in the field of survey design? Which of these two approaches an organisation settled on determined much of what followed in terms of the kinds of staff who were given responsibility for CAI policy and implementation. For example, on the key matter of choice of CAPI software, government and public sector organisations which saw CAPI as a computing matter (perhaps the majority) tended to favour writing their own bespoke software or to adopt CAPI packages which recognisably involved an approach similar to writing bespoke programs (these packages were often based on CATI software developed in an earlier computing generation). Commercial organisations were less concerned about a bespoke capability but their division of labour tended to leave even simple CAI packages as a matter for specialists. The government and public sector organisations which saw CAI as principally a matter for survey designers (researchers) tended to choose software which had essential functions built in (like the packages designed for commercial surveys, but with a much greater range of functions and a capacity to deal with complex data structures). The software had to demand no greater computing skills than their researchers already exercised in using statistical analysis packages like SPSS for manipulating data using logic as well as just running tables. To many such organisations, Blaise seemed to have been designed to meet their needs.

Many organisations saw CAI as essentially a computing development. This paper is about one of the organisations which saw it as essentially a development in survey design.

For such organisations, three main ways of working effectively have been identified:[vi]

- a team of researchers and programmers makes up an instrument design team for one or more surveys;
- researchers are trained as CAI programmers, and their skills kept fresh;
- researchers use standard modules and, where that is not possible, template modules to build questionnaires.

This paper describes why and how the author's organisation has used all three of these strategies since it began using CAI in production in 1990. It has moved increasingly towards the third strategy as the number of CAI surveys it carries out has expanded. The nature of the organisation is important for understanding the choice, so some brief details follow.

## Social Survey Division and Blaise

Social Survey Division of the UK's Office for National Statistics (SSD/ONS) comprises some 190 headquarters staff and 1,000 interviewers (including about 200 telephone interviewers who work in a central CATI unit). The interviewers are recruited on a permanent basis, following successful completion of stringent recruitment and selection procedures, training and a trial period of six months. In contrast to the norm for commercial agencies in the UK, they are paid for time and expenses rather than by completed interview, and very few do any work for other agencies.

The headquarters staff comprise nearly 80 who are involved in aspects of fieldwork management and training (Field); over 50 researchers who combine project management with survey design, analysis and report-writing or who work in a methodology unit; about 30 computing specialists; and over 20 staff who provide sampling, management and secretarial support. At any one time, the work of the division covers 9 continuous surveys, including an omnibus survey, and various stages of some 40 *ad hoc* surveys, and a range of methodological projects. Most, but not all, projects are household surveys, often involving interviews with all adults in the household. The two largest surveys together involve about half a million interviews with adults each year. Other surveys tend to be longer and more complex, and to involve smaller samples. Conventional enterprise surveys by mail or electronic means are carried out in a separate Business Statistics Group of ONS. The majority of SSD/ONS's work has been acquired through competitive tender. The surveys that SSD/ONS carries out are large-scale, usually complex, national studies for government and other public sector bodies. The uses of the survey results in designing and monitoring public policy make high quality essential. SSD/ONS clearly meets the criteria which were identified in the first paragraph of this paper for an organisation which has the basis for making successful use of CAI, and CAPI in particular.

Work is carried out in a project management structure, with each project team led by a researcher who is finally accountable for all aspects of the project. As it is researchers who write Blaise questionnaires in SSD/ONS, it is worth considering their characteristics more closely. Researchers are required to have a good (and, in practice now, often a higher) degree in a social science with substantial statistical content, statistics or mathematics; or to have demonstrated equivalent experience. Most are social scientists. While some have, or acquire, expertise in particular subject matter areas, it is SSD/ONS policy that researchers should be able to work on the full range of its surveys; in practice, movement between subject areas at the completion of each project, and between continuous and *ad hoc* surveys, is the norm in a researcher's career.

One result of the project management structure is that the researchers have always had hands-on knowledge and control of critical elements of the design and analysis, such as sampling, questionnaire and processing design, and statistical computing involved in the analysis. Under PAPI, there was already a tendency to try to eliminate duplication of tasks within the project team, such as specification of processing or analysis requirements to a programmer. This tendency was encouraged by software developments and by the need for programmers to take on more challenging tasks, to meet the needs of the organisation and their own careers.

As researchers had always designed paper questionnaires and were also seeking to extend their hands-on control of processing, it is perhaps not surprising that when CAPI became a real possibility in the mid-1980s SSD/ONS saw CAI questionnaire software as a tool for researchers. It quickly adopted Blaise because, in addition to its strong functionality and reliability, it was the only CAI software which had a fully integrated design. A single specification - in the familiar format of a questionnaire, though now incorporating edit checks - generated metadata which could be used unchanged from data collection (in any CAI mode) to analysis. That principle - signalled in the use of the general term *datamodel* for the enhanced questionnaire - was built on strongly in Blaise III and remains central to the appeal of Blaise to its users.

While it was important for researchers to take on direct specification of the instrument, it was equally important that they should not spend their time doing work for which they were not properly skilled and which it was more efficient for computing specialists to carry out. For example, SSD/ONS assigned the task of providing a suitable CAI environment of case management and telecommunications to its Survey Computing Branch.

In the early days of CAPI in SSD/ONS, there was a clear division of labour. At that time Blaise and Manipula were separate modules. Researchers used Blaise to write questionnaires and press-button Blaise utilities to output the data. If the survey was relatively simple, the output was read in SPSS and there was no need for any intervention by computing specialists. Usually, however, the survey would have a complex structure. In these cases, computing specialists read the output into their standard database software, Clipper, from where they could manipulate it into a suitable form for SPSS or, if necessary, send it to specialist software for handling hierarchical data. They did not need to know anything about Blaise: it was policy for them to treat Blaise as a "black box".

This division of labour was relatively easy to operate when only a few surveys had converted to CAI. The number of researchers who needed to learn to use Blaise was small, and the pioneers were highly motivated. Moreover, most of the early CAI surveys were continuous surveys carried out by small teams of researchers who could sustain some collective knowledge of Blaise so that they were not dependent on particular individuals. However, from 1994 (by when the continuous surveys had adopted CAI) SSD/ONS's policy was that any new survey would use CAI unless there was a very good reason why it should not (such as that it was a qualitative study involving large amounts of verbatim reporting). By 1995, over 95% of SSD/ONS's interviews were carried out in Blaise. Any and every researcher now needed to be able to write Blaise instruments or, in the case of senior managers, to understand enough of the critical design issues to be able to supervise junior staff who were doing the actual writing.

With the integration of Manipula and Blaise in Blaise III, there were efficiencies in the computing specialists using Manipula rather than Clipper for many purposes. Moreover, data output was more flexible than in the push-button utility days of Blaise 1 and 2, but it was also more of a file-specification task appropriate for computing specialists. Use of Manipula and Cameleon by computing specialists entailed their learning about Blaise datamodels. Although Blaise was no longer a "black box" for computing specialists - and, indeed, they were clearly far more skilled than researchers in purely computing aspects of Blaise (e.g. making its functions perform more than they were originally intended for; linking in external programs; understanding how Blaise metadata could be accessed for an automatic documentation tool written in Clipper) - the basic division of labour was retained. Researchers designed survey instruments directly in Blaise. Computing specialists output the data, using Manipula and Cameleon, and continued to provide and improve the case management and telecommunications environment for CAI data collection.

A management review confirmed the policy that the researchers working on a survey should write its Blaise instrument. The review took particular account of the importance for survey quality of hands-on control and knowledge of the questionnaire by researchers. It also stressed the importance of co-operation in the survey team between researchers and computing specialists to ensure that the requirements for efficient output design were built into the datamodel by the researchers from the start. The review regarded the mode of working which involved researchers writing specifications for programmers to put into Blaise as involving duplication of resources and as potentially error-prone. It noted the difficulties reported by some organisations in recruiting and retaining programmers to carry out such (for them) mundane work[vii]. Finally, the review recognised that researchers would spend only a small and intermittent part of their careers writing Blaise questionnaires. This meant that there was a particular need for a cost-effective mechanism to ensure that researchers had the necessary up-to-date skills at the point that they were required; and that all Blaise questionnaires were of high quality. The remainder of this paper describes the mechanism that was set up.

## Standards and quality assurance

The problem is to provide more than 40 researchers with a sufficient knowledge of Blaise for them to be able to use it as a tool to write high quality survey instruments, without undue cost or time in training and keeping the skills up to date. They should spend no more time than they do in, say, keeping up to date with analysis software. There are two main types of requirement for knowledge about Blaise. Junior researchers are likely to write all or most of the Blaise instrument code on their projects. Senior researchers need to know enough of Blaise and the critical factors in design using Blaise to be able to supervise the juniors. The more senior the researcher the less need there may be to know the details of Blaise, but at the very least there is a need to know what may affect management decisions and strategy.

After nearly a decade of Blaise use in SSD/ONS, many senior researchers have experience in writing Blaise code from their junior days. However, it may some time since they last used Blaise. Other senior researchers do not even have this experience to build on. Brief guidance on "Blaise for managers" has proved sufficient for most senior managers if they have some former or background knowledge and will not need to write code themselves.

New researchers are recruited at all levels, though in largest numbers at the junior level. New recruits receive a 3-day course in Blaise, provided by Statistics Netherlands. This is enough to enable them to start to understand the complexities of real survey instruments, but there is a great deal to learn about Blaise as a tool in survey design and not simply as a programming language. Such skills are only learned in practice, so more courses are not the answer. At the same time, SSD/ONS has to provide secure means of delivering high quality Blaise instruments to carry out customers' surveys.

The strategy SSD/ONS has adopted is to embody the collective practical knowledge of the organisation in:

- a set of standards for writing Blaise code;
- standard code for common elements of questionnaires, such as the household box;
- and templates for writing unique questionnaire content into standard instrument structures.

Annex A is an extract from the document which explains this strategy and some of its detailed implementation to SSD/ONS's researchers.

Among its other benefits, the strategy simplifies the training requirement. Instead of learning and re-learning Blaise each time it is needed, the researcher needs to understand only some basic structural principles and conventions which can be summarised in a few pages. All the detail, in its most up-to-date form, using best practice, is available in the standard code and templates.

There are other significant advantages to this approach:

- supervision of new and junior researchers is simple and efficient, since it is clear to them and to their supervisors exactly what they should produce;
- researchers can read each other's code with minimal difficulty since it looks like the code they themselves would write, and uses the same conventions; this is important in an environment where researchers move between surveys and where several researchers will work together on the larger surveys;
- the screens which interviewers see have a standard format - they know, for example, where on the screen to find the current respondent's name (and it is guaranteed that the researcher will not forget to provide such standard information);
- the approach facilitates rapid development of reliable instruments by allowing researchers to build them, to a considerable extent, from standard blocks;
- it facilitates testing and SSD/ONS's quality assurance of its Blaise instruments.

The strategy requires a small team to lead, co-ordinate and publicise the setting of standards and the provision of quality assurance. The team, which is called the CAI Standards and Quality Assurance (SQA) team, comprises two researchers, a fieldwork expert and two computing specialists. Their jobs on the team are part-time, and the doubling-up of researchers and computing specialists is to ensure cover. The total time allocated to the team, to cover all its members, is well under one person year and is tending to reduce. In the allocated time, the team develops new standard and model code, provides a trouble-shooting service (advice to, but not substantive work on, survey projects) and maintains its own expertise. The computing specialists check all new instruments for conformity to the standards, mainly by using automatic tools. The quality of the content, and its testing, remains the responsibility of the senior researcher on each survey.

Conforming to standards does not add time or cost to a researcher's task: it is the default option, embodied in the standard and model code that researchers start from. The standards would never be allowed to prevent a researcher from designing an instrument to function as they and their customers choose. In practice, the issue does not arise. Blaise needs to be written in some form, and the model code embodies a suitable one. It is easier to use the standard and model code than to invent a form of one’s own.

The approach described in this paper has grown up over many years. The SQA team has only formally existed for two years, but there has been a CAI co-ordination role from the start. The approach has been re-examined critically several times over the years, but has always been reaffirmed as the best way to meet SSD/ONS’s goals in survey design.

# ANNEX A

### STANDARDS AND CONVENTIONS FOR WRITING BLAISE III DATAMODELS IN SSD/ONS

***Standards and Quality Assurance (SQA) team***

## Blaise III standards (document month: September 1997)

A standardised approach to CAI instrument design is one of SSD's strategies for helping designers produce instruments which meet the needs of their surveys as fast and as accurately as possible. It is the basis for preprepared modules of Blaise code (e.g. for harmonised questions) which can be slotted into any SSD survey, and for templates of Blaise code which can be adapted to any samples and subject matter. Standard modules and templates will save designers a lot of tedious and error-prone work which is associated with any programming task and allow them to concentrate on the real research issues. They will help interviewers by prompting designers to produce screen layouts with the information they need in the places where they expect to find it and with standard meanings. Many of the standards in this document are essential to the smooth running of survey instruments in the laptop and CATI computing environments. Standardisation is a means for an organisation, and not just individuals, to learn from experience and to generalise good practice.

….
The SQA team has drawn up the standards. It does not claim that these standards are the only viable way of doing things, but asks people to accept that the point of standardisation is to have a common way wherever possible and that these particular choices are based on experience of a wide range of surveys and of the problems raised by the differing styles of their designers. The process of instrument design will work best if the survey team meets with the SQA team before actually writing any code and agrees a strategy for the design, including the way standards will be used, and arranges any help that the SQA team can provide. Many surveys have already done this successfully ….

This document is for quick reference for designers who have at least a basic knowledge of Blaise. It is about SSD's standards for Blaise, not about any requirements of the software itself.

### *Questionnaire ("Datamodel"[7]) structure*

A survey datamodel will comprise:

- one and only one datamodel control file (with a file extension .bla) (see Annex A.1) covering all requirements at datamodel level except the following paragraphs:
  RULES
  TYPE
  LOCALS
  AUXFIELDS

In the main, it will be a list of INCLUDE files, to bring in the paragraphs excluded above and the blocks for the survey. These INCLUDE files will have extensions .HAR (harmonised, not to be changed except with written agreement of SQA team) or (survey-three-letter-acronym (TLA) for example .GHS). In addition to the harmonised blocks, the SQA team will provide model files (extension .MOD) for researchers to use as templates for their own survey blocks (converting the extensions from .MOD to .(survey acronym); in the rest of this note, we refer to the modifiable files as .MOD but a survey will need to change this extension).

- INCLUDE files for the following datamodel level paragraphs:
  RULES......................RULES.MOD (see example at Annex A.2)
  TYPE........................TYPE.HAR and TYPE.MOD
  LOCALS...................LOCALS.HAR and LOCALS.MOD
  AUXFIELDS............AUXFIELDS.HAR and AUXFIELDS.MOD

- INCLUDE files for the survey's blocks (ONLY 1 block or 1 block which contains nested blocks per file):
  harmonised blocks..... ...................QSubject.HAR
  model/modifiable....... ...................QSubject.MOD

It is very important to note that ONLY 1 block or 1 block which contains nested blocks is allowed per file. This is vital for efficient instrument design. For example, it allows the naming convention of the file and block having identical names, which facilitates instrument amendment (see below).
….

Blaise III does not allow CHECKS and SIGNALS to be separated off into their own INCLUDE files. This Blaise constraint means that the RULES paragraph at datamodel level is not, as we would like, just an outline of the routing. The CHECKS and SIGNALS should be put in carefully, in as organised and discrete a way as possible, so as not to clutter up the view of the routing. The model code RULES.MOD may be used as a template.

### *Naming/defining conventions*

Datamodel control file........................survey acronym + surveyYEAR + version .BLA

[7] This Blaise term is intended to indicate that the specification for the questionnaire is much more than that - it is the metadata (data about the data) for the survey, used at all stages from data collection to analysis.

(e.g. SEH9601A.BLA, OMN9702B.BLA)

INCLUDE files exactly same as field name of block in each
(e.g. QTCalls is in QTCALLS.MOD; the rules for the datamodel are in RULES.MOD). This means that names of blocks should be no more than 8 characters (the DOS limit) even though Blaise allows more.

Blocks .................................. start with B (e.g. BPerson)
.................................. field name of block starts with Q (e.g. QPerson)
.................................. table blocks start with T (e.g. TPerson)
.................................. field name of table block includes the T (e.g. QTPerson)
.................................. Block field names are always defined in the INCLUDE file they belong to, NOT in the datamodel FIELDS paragraph.

Locals .................................. with 1 or 2 standard exceptions ( the current list is at ..), names will comprise the name of the block in which they are defined, with L in front
and a number at the end (e.g. LPerson1, LPerson2 would be the ones defined in BPerson; LTPerson1 would be found to be defined in TPerson). Datamodel level locals will be called LDM1, LDM2, etc.
(Reason: so-called intuitive names are frequently misleading or not actually intuitive for another reader. What one needs to know is where the local is defined. Then the reader can find out what it's actually doing.)

Derived variables ................................ Computed variables stored on file (i.e. computed into fields) must start DV

Paragraph commands ......................... FIELDS, AUXFIELDS etc - must ALWAYS be accompanied by a commented reference to the block they are in, e.g. RULES { TPerson}

Labels............................................. Labels can be used to provide precise comments which have a reasonable chance of being updated (i.e. it's a project responsibility) since they are part of the survey metadata,
e.g. on blocks
Block BPerson “Main person Information Block”

……..

### *Standard files and blocks*

….details of standard configurations to ensure that testing on office PCs mimics accurately the configurations on interviewers’ laptops……

(an example…..)

### *Permission to recall block (QRecall.HAR)*

There is a standard block for asking for permission to recall. Your datamodel control file should INCLUDE it as follows:

INCLUDE O:\B3115\SQA\MODULES\QRecall.HAR

This guarantees that you will use the latest version. If you wish to use something else, you should have written agreement from the SQA team.

## *Annex A.1 - Datamodel control file example.*

```
{TM020297  **********************************************************
* GLC97nn - DATAMODEL CONTROL FILE for GENERAL LIVING CONDITIONS SURVEY
* 1997 (the Blaise III Field training questionnaire)

* GLC97nn.BLA - this file - contains INCLUDE files for whole
* questionnaire.
*
* It does NOT contain TYPES, LOCALS, AUXFIELDS or RULES - see
* include files TYPE.HAR, TYPE.GLC, LOCALS.HAR, AUXFIELD.HAR,
* AUXFIELD.GLC and RULES.GLC respectively.
*
* RULES.GLC must be the final include file before ENDMODEL.
TM020297      **********************************************************}


DATAMODEL GLC9704

LANGUAGES = ENG "English", HLP "Instructions"

ATTRIBUTES = DontKnow, Refusal

PRIMARY

  QID

PARALLEL

  QReCall
  QHAdmin

USES { DM ***********************************************************}

{TM301296*************************
* check for valid SOC codes
TM301296}

  SocMeta 'C:\CASEBOOK\COMMON\EXTSOC',

{TM301296*************************
* classification matrix
TM301296}

  MatMeta 'C:\CASEBOOK\COMMON\EXTMAT',

{TM301296*************************
* subject of qualifications - from LFS - see also LIBRARIES para (note
* slightly different name there: Subjects (plural)
TM301296}

  Subject 'C:\CASEBOOK\GLC\SUBJECT',

{TM301296*************************
* nationality - from LFS
TM301296}

  Nation 'C:\CASEBOOK\GLC\NATION'

{TM301296*************************
* END OF USES {DM} PARAGRAPH
```

```
TM301296************************************************************}

LIBRARIES { DM ***********************************************************}

{TM200296*************************
* subject of qualifications - from LFS - see also USES para (note
* slightly different name there: Subject (singular)
TM200296}

  Subjects 'C:\CASEBOOK\GLC\SUBJECTS'

{TM200296*************************
* END OF LIBRARIES {DM} PARAGRAPH
TM200296************************************************************}


INCLUDE "QID.HAR"
INCLUDE "QDATABAG.GLC"
INCLUDE "TYPE.HAR"
INCLUDE "TYPE.GLC"
INCLUDE "LOCALS.HAR"
INCLUDE "LOCALS.GLC"
INCLUDE "AUXFIELD.HAR"
INCLUDE "AUXFIELD.GLC"
INCLUDE "QSIGNIN.HAR"
INCLUDE "QNAMES.HAR"
INCLUDE "QTHCOMP.HAR"
INCLUDE "QHOH.GLC"
..
..
INCLUDE "QTENURE.HAR"
INCLUDE "QECSITU.GLC"

{TM080197
START OF ADULT INCLUDE FILES
TM080197}

INCLUDE "QTISTART.HAR"
INCLUDE "QTILO.HAR"
INCLUDE "QTWANTJB.GLC"
INCLUDE "QTLASTJB.GLC"
INCLUDE "QTMAINJB.HAR"
INCLUDE "QTEMPLEN.HAR"
..
..
INCLUDE "QTINCEMP.GLC"
INCLUDE "QTINCSE.GLC"

{tm200697  ***********************************************************
* The next file "QRECALL.HAR" asks for permission to recall
* and collects contact details
TM200697}

INCLUDE "QRECALL.HAR"

{tm010197  ***********************************************************
* The next file "QHADMIN.(surveyname)" is the one that Social Survey
* Division calls the "ADMIN BLOCK".
* It contains 3 INCLUDE files: "QFAMUNIT.HAR","QTOCC.HAR" and
* "QTCALLS.HAR"
TM010197}

INCLUDE "QHADMIN.HAR"
```

```
{tm160197  **********************************************************
* The next file "RULES.(surveyname)" must be present and must be the last
* include file.
*
* "RULES.(surveyname)" contains the 2 datamodel level CHECK
* paragraphs - (1) harmonised (2) model (i.e. survey-specific).
* Note that these CHECK paragraphs cannot be held in INCLUDE files.
tm160197}

INCLUDE "RULES.GLC"

ENDMODEL
```

### *Annex A.2 - Example of a file containing a block.*

```
{TM010197
* This file (QNames.HAR) sets up an array in which the names or other
* identifiers for household members can be entered in any order, in response
* to question "Who lives here?" (asked in QSignIn).  Household reference
* person, if needed, can be identified later, in a question or by a compute.
*
* ONLY ONE THING CAN BE ALTERED IN THIS FILE - AND ONLY WITH AGREEMENT OF
* STANDARDS AND QA TEAM - (THE ARRAY SIZE AT …)
tm010197}

BLOCK BNames

  LOCALS { BNames }

     LNames1, LNames2 :  INTEGER

  BLOCK BBNames

     FIELDS { BBNames }

        Name     "RECORD THE NAME (OR A UNIQUE IDENTIFIER) FOR HOH,
                 THEN A NAME/IDENTIFIEER FOR EACH MEMBER OF THE
                 HOUSEHOLDúúúúúHELP<F9>@/@/
                 WHEN ALL HOUSEHOLD MEMBERS HAVE BEEN ENTERED, PRESS
                 PgDn@/"

                 "PRIMARY SET OF QUESTIONS ON HOUSEHOLD COMPOSITION
                 AND RELATIONSHIPS ASKED ON ALL SURVEYS.@/@/
                 HOUSEHOLD COMPOSITION@/@/
                 Stage 1: Establish Residency - only/main residence 6
                 month rule/ Check Adult Children@/
                 Stage 2: One or more households - 'Do you all share
                 at least one main meal a day or share living
                 accommodation?'@/
                 Stage 3: Establish HoH - 'In whose name is the
                 accommodation owned or rented?'@/"

                 : STRING[12],EMPTY

     RULES { BBNames }

        Name

  ENDBLOCK { BBNames }

  FIELDS { BNames }

     QBNames  :  ARRAY [ 1..10 ] OF BBNames
     DVHsize    :  1..10,EMPTY

  RULES { BNames }

     LNames2  := 0
     FOR LNames1 := 1 TO 10 DO
        QBNames[LNames1]
        IF QBNames[LNames1] <> EMPTY THEN
           LNames2 := LNames2 + 1
        ENDIF
     ENDDO
     DVHsize  := LNames2
```

```
     DVHsize.KEEP

ENDBLOCK { BNames }

FIELDS  { DM }

  QNames :  BNames
```

---

[i] A recent summary of the state of the art in CAI is to be found in Couper, M. et al (eds) *Computer Assisted Survey Information Collection*, 1999, John Wiley & Sons Inc, New York.

[ii] See in this volume: Wings, H. *Blaise for the Internet.*

[iii] *The History and Development of Computer Assisted Survey Information Collection Methods*, Couper, M.P. & Nicholls, W.L. in M.P. Couper et al. (eds) 1998

[iv] *Development and Implementation of CASIC in Government Statistical Agencies*, Clark, C.Z.F., Martin, J. and Bates, N. in M.P. Couper et al. (eds) 1998

[v] *Integrating CASIC into Existing Designs and Organizations: A Survey of the Field*, Groves, R.M. & tortora, R.D. in M.P. Couper et al. (eds) 1998

[vi] *Producing CAI Instruments for a Program of Surveys*, Pierzchala, M. & Manners, T. in M.P. Couper et al. (eds) 1998

[vii] *Computer Aided Interviewing: Has It Ever, and Will It Still Work?* Connett, W.E. in Association for Survey Computing, Proceedings of The Second ASC International Conference, London,1996