CHAPTER 7

PROCESSING THE DATA

This chapter is written for survey coordinators, data processing experts and technical resource persons. It provides information on how to:

- Prepare for processing the data
- Set up a system for managing data processing
- Carry out data entry
- Edit the data and create a 'clean' data file for analysis
- Produce tabulations with the indicators
- Archive and distribute data

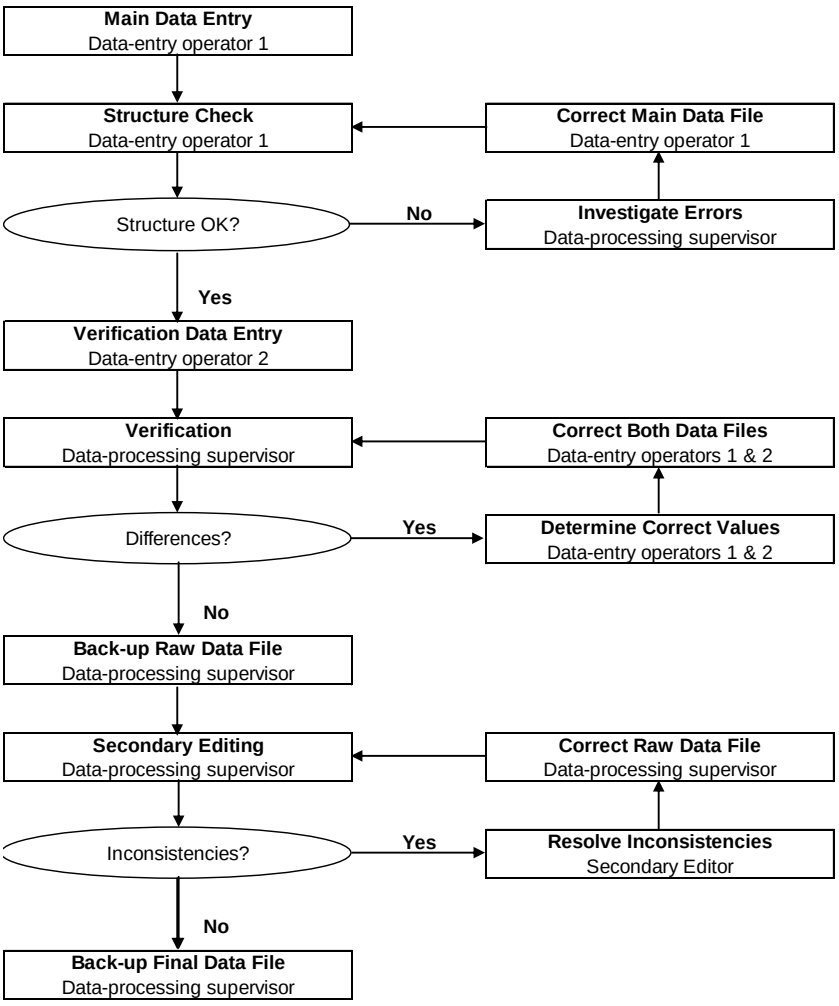
The MICS3 data-processing system is designed to deliver the first results of a survey within a few weeks after the end of fieldwork. This chapter contains information that will help you to undertake the planning and advance preparation that will make this goal a reality. The chapter begins by giving you an overview of the MICS3 data-processing system. It then discusses each of its components in detail, providing references to supplemental sources of information where appropriate. It closes with a set of three checklists that will help you make the processing of your survey data a success.

OVERVIEW

The reason that the MICS3 data-processing system can achieve such rapid turnaround time is because data is processed in tandem with survey fieldwork. Data for each cluster is stored in a separate data file and is processed as soon as the questionnaires are returned from the field. This approach breaks data processing down into discrete segments and allows it to progress while fieldwork is ongoing. Thus, by the time the last questionnaires are finished and returned to headquarters, most of the data have already been processed.

Processing the data by clusters is not difficult, but it does require meticulous organization. The data-processing system can be divided into three phases: preparation, primary data processing and secondary data processing. Each of these phases is summarized in the sections that follow, and each has its own checklist at the end of the chapter.

Table 7.1
The MICS3 Data Processing System



PREPARATION FOR DATA ENTRY

The goal of preparing for the data-entry phase is to be ready to begin shortly after the fieldwork commences. The preparation phase involves the following steps:

- Obtaining computer equipment and setting up a data-processing room
- Identifying and recruiting appropriate personnel
- Adapting computer programs to the country-specific questionnaire
- Setting up a system for managing the questionnaires and data files.

PRIMARY DATA PROCESSING

The goal of primary data processing is to produce clean, edited data files. Primary data processing involves the following steps:

- Entering all questionnaires for a cluster onto a data file
- Checking the structure of the data file
- Entering the data a second time and then verifying the data file
- Backing up the checked and verified data file
- Performing secondary editing on the data file
- Backing up the edited, or final, data file.

The flow of primary data processing is summarized in the flow chart on the previous page. Note carefully that structure checking, the verification of data entry and secondary editing are iterative procedures that are repeated until all problems are resolved or determined to be acceptable.

SECONDARY DATA PROCESSING

The goal of secondary data processing is to produce analysis data files and to create the MICS3 standard tables. Secondary data processing involves the following steps:

- Concatenating all cluster data files into one data file
- Exporting the data to the SPSS software
- Calculating sample weights
- Computing wealth index
- Recoding variables to simplify analysis
- Creating the tables required to analyse the data
- Archiving and distributing the data files.

PERSONNEL AND INFRASTRUCTURE

PERSONNEL

The data-processing team for a MICS3 survey includes four types of personnel: a questionnaire administrator, data-entry operators, secondary editors and a data-processing supervisor. Each position has distinct responsibilities and combining them is likely to damage the quality of your data.

The **questionnaire administrator** (or **office editor**) checks and organizes questionnaires as they arrive from the field. When a cluster arrives at the data-processing office, he/she checks that all of the questionnaires are present and ready to be entered. If there are missing questionnaires, he/she must resolve the problem with the help of the fieldwork team (the precise steps that the questionnaire administrator must take are detailed later).

The **data-entry operators** enter the data. They should have prior data-entry experience and be familiar with the questionnaires. One way to accomplish this is to have the data-entry operators attend the interviewers' training. Before beginning data entry, a separate 2-3 day training session must be held to acquaint data-entry operators with the data-entry program and the rhythm of the data-processing system. By the end of the training, the data-entry operators should be comfortable with the data-entry program and aware of their daily responsibilities. The required number of data-entry operators depends upon the number of available computers and is discussed in detail below.

The **secondary editors** investigate and resolve complex inconsistencies discovered by the secondary editing program. They must have an excellent understanding of the questionnaires and the goals of the survey. Editing guidelines are provided in Appendix Seven to aid them in the secondary editing process. A typical survey will require one or two secondary editors.

The **data-processing supervisor** is a critical member of the data-processing team. He/she adapts the model programs to suit her/his country's questionnaires and oversees all data-processing tasks. The data-processing supervisor should have experience managing data processing for a large-scale survey or census, an excellent understanding of the questionnaire, and programming skills in the CSPro and SPSS software packages. The data-processing supervisor should be available on a full-time basis during the period that the data are being entered, edited and tabulated.

The data-processing supervisor should be identified early in the planning stages of the survey so that he/she can be involved in the revision of the MICS3 questionnaire. This person should be consulted to ensure that the coding schemes used in the questionnaire are consistent and unambiguous and that all of the identification information needed is included. The data-processing supervisor must also be able to assist in final revisions to the questionnaire based on experience gained while entering questionnaires from the pre-test.

COMPUTER EQUIPMENT AND OTHER HARDWARE

Below is a list of equipment necessary for data processing:

- Data-entry computers
- The data-processing supervisor's computer
- A secondary storage device (for example, a portable USB device or CD-RW drive)
- Diskettes (or a means for operators to transfer files to the data-processing supervisor, for example, via a network)
- A printer
- Paper
- Toner cartridges/printer ribbons
- Surge protectors
- Uninterruptible power supplies (UPS)
- Green pens

The data-entry computers should have Pentium processors, Windows 95 or higher, at least 32 megabytes of RAM, 1 gigabyte or more of free hard-disk space, and 3.5-inch floppy diskette drives (or be networked together). The number of data-entry computers needed to process the survey depends on the size of the sample, the number of hours a data-entry operator will work each week, the space available and the timetable for the survey. To obtain an estimate of the number of computers needed for data entry, you should estimate how long a data-entry operator will need to enter the questionnaires for a typical household and multiply that by the number of households expected, according to the sample design. If you aren't able to estimate the time it will take to enter the questionnaires, use 20-30 minutes per household as a rough guide, depending on the number of women and children expected per household. Multiply this time estimate by the number of households to get the total hours needed for data entry. Divide this estimate by the number of hours each operator will work per week and then by the number of weeks you will have to complete data entry (if you aim to complete it within a week after the last questionnaires have been returned from the field).

For example, if the sample size for the survey is 6,000 households and each household takes 20 minutes to enter, the total time needed to enter all of the households is 2,000 hours. If you have 8 weeks to complete the data entry, then 250 hours per week are needed. If each data-entry operator works 40 hours a week, you will need seven computers and seven data-entry operators. Sometimes it is possible to arrange double shifts of data entry so that one computer is used by two data-entry operators each day. Each operator would work for, say, 6 hours, so that the computer is in use for 12 hours a day.

The supervisor's computer should have a faster processor, Windows 98 or higher, at least 64 megabytes of RAM, 1 gigabyte or more of free hard disk space, a 3.5-inch floppy diskette drive (or be networked to the data-entry computers), and a secondary storage device.

Uninterruptible power supplies and surge protectors are essential if the country in which you are working suffers from power outages. Green pens should be used whenever a member of the data-processing team modifies the data on a questionnaire. The green ink distinguishes these changes from the original data recorded by the interviewer (in blue ink) and any changes made by the fieldwork team (in red ink).

SOFTWARE PACKAGES

The standard programs for processing MICS3 surveys were developed in CSPro 2.6 and SPSS. CSPro, which has been used to process both surveys and censuses, was developed collaboratively by the United States Census Bureau, ORC Macro International and SerPro Ltda. It can be downloaded free of charge from the website of the US Bureau of Census.[1] SPSS is a commercial software package that is available through UNICEF and through many software suppliers.

OFFICE SPACE

Separate rooms are required for data entry and data editing. The data-entry room should be large enough so that each data-entry operator has space for her/his computer and the questionnaire on which he/she is working. There should be desks or tables for working and sufficient electrical outlets. The room should be cool, well lit and as free from dust and humidity as possible. In countries with hot climates, this requires that the room be air-conditioned. An uninterruptible power supply should be connected to each computer. If power outages are likely to be frequent or prolonged, another emergency power supply, such as a generator, is necessary.

The data-editing room is for the questionnaire administrator and the secondary editors. It, too, should be cool and well lit, and there should be sufficient space for the editors to review questionnaires. Ideally, the editing room will contain sufficient shelves or cupboards to store the questionnaires in an organized fashion. If the questionnaires cannot be stored in the editing room, then they should be stored nearby and be easily accessible since they will be needed at various stages throughout processing. Be careful not to underestimate the amount of space that will be needed to store the thousands of questionnaires that you will have in the office by the end of the fieldwork.

1 The web address is: http://www.census.gov/ipc/www/cspro/

ADAPTING THE STANDARD PROGRAMS

As outlined in Chapter 3, the model MICS3 questionnaire must be adapted to the situation in each country. This means that the model data-entry, editing and tabulation programs must also be modified to be consistent with the changes made in the questionnaire. The more changes that are made to the model questionnaire, the more time must be allocated for adapting and testing the programs. For example, if new questions are added to the questionnaire, corresponding additions must be made in the data-entry, editing and tabulation programs.

This process will be significantly easier if the question numbering in the model questionnaire is maintained. If questions are added, a letter should be added to the existing numbering (for example, a question inserted between WS4 and WS5 should be numbered WS4A). Similarly, if questions are deleted, the remaining questions should not be renumbered. In addition, when coding categories are added to those in the model questionnaire, they should be added to the end of the existing list, leaving the other codes intact. The adaptation of data-entry and editing programs should be completed prior to the pre-test. Questionnaires from the pre-test can be entered and edited using the programs. Following these instructions will serve two purposes. It will reveal problems in the coding and skip patterns in the questionnaires as well as any errors in the programs. Once the pre-test has been completed and the questionnaire finalized, final changes can be made to the programs. Subsequent sections give basic guidance on modifying the model data dictionaries and the model CSPro applications. A more detailed summary of the contents of the CSPro applications is provided in separate documents.

Even if you are not adding questions to the model questionnaires, the model data dictionaries and applications contain certain items that must be updated (for example, the acceptable range for the date of interview, the acceptable range for the cluster number, etc.). These items are necessarily country-specific and must be completed by you. Thus, even if your country uses the model questionnaire, you will have to adapt the standard programs.

THE DATA DICTIONARIES

In the Multiple Indicator Cluster Survey, groups of related questions (for example, on maternal mortality, contraceptive use and immunization) are collected into modules that are then collected into questionnaires (that is, for the household, for individual women and for children under five). In CSPro, dictionaries are used to describe this data structure: a group of related variables (questions) comprises a record (module), and a group of records comprises a level (questionnaire). These are stored in a dictionary file (extension: *dcf*). In addition to the data dictionary, forms linked to the dictionary are used for data entry. There is usually one form for each record. The forms are stored in a forms file (extension: *fmf*). The *dcf* and *fmf* files can be modified directly. The best way to do this is to open the forms file in CSPro. This will give you access to the data dictionary and the forms together and ensure that the two remain synchronized. It is advisable to keep a back-up of the model data dictionary and forms file for reference.

There are three types of MICS3 questionnaires. The Questionnaire for Individual Women and the Questionnaire for Children Under Five correspond to a single unit of analysis: a woman and a child, respectively. The Household Questionnaire contains two units of analysis: the household and the household members. All of the questionnaire types are stored in *mics3.dcf* and *entry.fmf*.

IDENTIFICATION VARIABLES AND LEVELS

In CSPro, every questionnaire must have a series of variables that uniquely identifies it. For example, a household is identified by its cluster number and household number. The variables that identify a questionnaire are known as the identification variables. Table 7.2 below lists the questionnaire types and their identification variables.

Table 7.2
Questionnaire Types and Their Identification Variables

Questionnaire	Cluster number	Household number	Line number
Household	HH1	HH2	
Women	HH1	HH2	LN
Children	HH1	HH2	LN

As you can see from the table, women and children have the same identification variables. Since each household member is listed on a separate line in the household listing, no two women or children will have the same line number, even if they are in the same household. Thus, combined with cluster number and household number, line number uniquely identifies a woman or child.

In a CSPro dictionary, a level is defined by a set of identification variables. In the MICS3 dictionary, there are two levels: households and individuals (that is, eligible women and eligible children). Households are the first level while women and children are the second level. This hierarchical structure is natural since in the MICS3 questionnaire every woman or child belongs to a household while a given household may have many women and children.

The women's questionnaire and children's questionnaire are stored on the same level because each applies to a household member. The data-entry application contains logic that skips forms pertaining to children when entering a woman's questionnaire and skips forms pertaining to women when entering a child's questionnaire. Thus, although women's questionnaires and children's questionnaires are both stored as level-two cases, they have no common variables except the identification variables.

MODULES

The data dictionary was designed to reflect the modular structure of the MICS3 questionnaires. Each module is stored in its own record (exception: the Household Listing and Children Orphaned and Made Vulnerable by HIV/AIDS modules each have two records because of their unusual structure) in *mics3.dcf* and each record has a form (or two, in the case of the Household Characteristics module) associated with it in *entry.fmf*. Thus, if your country does not use a particular module, you can remove it by deleting its record and its form (and removing any extra logic that references it from the data-entry application).

The modules available for the Household Questionnaire (with the module's code(s) listed in parentheses) are: Household Information (HH), Household Listing (HL and TO), Education (ED), Water and Sanitation (WS), Household Characteristics (HC), Insecticide-treated Nets (TN), Children Orphaned and Made Vulnerable by HIV/AIDS (OV and OR), Child Labour (CL), Child Discipline (CD), Disability (DA), Maternal Mortality (MM) and Salt Iodization (SI).

The modules available for the Questionnaire for Individual Women are: Women's Information Panel (WM), Child Mortality (CM), Tetanus Toxoid (TT), Maternal and Newborn Health (MN), Marriage/Union (MA), Security of Tenure and Durability of Housing (ST), Contraception (CP), Female Genital Mutilation/Cutting (FG), Attitudes Towards Domestic Violence (DV), Sexual Behaviour (SB) and HIV/AIDS (HA).

The modules available for the Questionnaire for Children Under Five are: Under-Five Child Information Panel (UF), Birth Registration and Early Learning (BR), Child Development (CE), Vitamin A (VA), Breastfeeding (BF), Care of Illness (CA), Malaria (ML), Immunization (IM) and Anthropometry (AN).

VARIABLE NAMING CONVENTIONS

Variables are named for the questionnaire module in which they are located and the number of the question whose response they contain. For example, question 9 in the Household Listing is stored in a variable named HL9. Some questions are split into two or more parts, with the separate parts identified by a unique letter. Each part of such questions is stored in a separate variable. The names of these separate variables include the letters that distinguish the parts of the question. For example, question 11 of the Maternal and Newborn Health module has two parts. The first part of this question is stored in the variable *MN11A*, and the second part is stored in *MN11*.

Some questions have two or more parts to the response categories. These questions are stored in a single variable and the response categories are defined as sub-items. When these questions concern dates, the letters 'd' (for day), 'm' (for month) and 'y' (for year) are appended to the base variable's name to create the name of the sub-items. In question 6 of the Women's

Information Panel, for example, the woman's day, month and year of birth are required. Her response is stored in *WM8*, which has three sub-items: *wm8d*, *wm8m* and *wm8y*.

Some questions have a structure in which the first part of the response is the form of the response and the second part is the response. These questions are stored in a single variable and the form and response are defined as sub-items. The name of the sub-item storing the form of the response is the name of the variable with the letter '*u'* (for units) appended to it, while the name of the sub-item storing the response is the name of variable with the letter '*n'* (for number) appended to it. For example, question 13 in the Maternal and Newborn Health module records how long after birth a child was first given breastmilk. The respondent may answer in hours or days. The response is stored in the variable *mn13* with sub-items *mn13u* and *mn13n*.

MULTIPLE RESPONSE QUESTIONS AND ALPHANUMERIC VARIABLES

There are a number of questions that allow for multiple responses. These questions are distinguished on the questionnaire by alphanumeric response codes (that is, the letters A through Z). In the data dictionary, the response to a multiple response question is stored in an alphanumeric variable whose length equals the maximum number of potential responses. These are the only alphanumeric variables in the dictionary. Each alphanumeric variable has one sub-item for each response code on the questionnaire. The name of one of these sub-items is the variable's name plus the response code that sub-item represents.

For example, the second question in the Maternal and Newborn Health module records all of the individuals from whom a woman received antenatal care before her last birth. The potential response codes are A, B, C, F, G, H, X and Y. The variable *mn2* is therefore eight characters long and there are eight sub-items: *mn2a*, *mn2b*, *mn2c*, *mn2f*, *mn2g*, *mn2h*, *mn2x* and *mn2y*.

CODING CONVENTIONS

The model dictionaries use standard coding for certain responses. We will first discuss coding conventions for numeric variables. The response 'Other' is always coded as a 6 with leading 9s. Inconsistent responses are always coded as a 7 with leading 9s. The response 'Doesn't know' is always coded as an 8 with leading 9s. Questions with a missing response (that is, the interviewer did not record a response to an applicable question) are always coded as a 9 with leading 9s. Questions that are not applicable to a respondent are always coded as a blank. Table 7.3 below summarizes the standard coding conventions.

Table 7.3
Summary of Standard Coding Conventions

Response	Variable length				
	Alphabetic	One character	Two characters	Three characters	Four characters
Other	X	6	96	996	9996
No/None	Y	Na	Na	na	Na
Inconsistent	na	7	97	997	9997
Doesn't know	Z	8	98	998	9998
Missing	?	9	99	999	9999
Not applicable	Blank	Blank	Blank	Blank	Blank

Because the codes 6 through 9 are reserved for special use, any question that requires more than six response categories should have 2-digit response categories with leading zeros (for example, 01, 02, 03, 04, 05, 06, 07, 96, 97, 98 and 99).

For alphanumeric variables, the response 'Other' is always coded as X, the response 'Doesn't know' is always coded as Z, a missing value is always coded using the question mark character (?), and not applicable is coded as a blank.

RANGES

Most of the questions in the MICS3 questionnaires have defined response ranges. The ranges are defined for variables in the dictionary *mics3.dcf*. CSPro checks during data entry that any value entered in a variable is within that variable's defined ranges. CSPro allows for a large number of ranges for each variable, so questions with non-consecutive response ranges (for example, 1-8, 96, 98 and 99) should be defined using several ranges (for example, 1-6, 96, 98 and 99, instead of 1-99). While dictionary ranges are useful for checking simple ranges, more complicated or conditional ranges (for example, consistency between day and month in a date variable) should be checked in the data-entry or editing applications.

THE DATA-ENTRY APPLICATION

The data-entry application is a long and complex program. Space limitations prevent it from being described in any detail in this chapter. Instead, this section will concentrate on some important general issues about data-entry application.

SKIPS

The MICS3 questionnaires make abundant use of skips. Skips are instructions on the questionnaire that tell the interviewer to skip all the questions between the current question and a question later on in the questionnaire. Skips on a questionnaire must be matched by skips in the

corresponding data-entry program. Skips in a data-entry program define the data-entry path. CSPro strictly enforces the data-entry path whenever the '*skip to'* or '*skip to next'* commands are used.

ERROR MESSAGES

If a data-entry operator enters a value for a variable that is inconsistent with previously entered information, it is useful to display an error message. This error message should explain the nature of the problem and provide any information that might help resolve the inconsistency. In CSPro, the *errmsg* function displays an error message with user-defined text whenever it is called. The error messages for the data-entry program are numbered and stored in the file *entry.mgf*. The text, number and inconsistencies that lead to each of these messages being displayed are listed in Appendix Seven, as are guidelines for resolving them.

You should review your questionnaire to determine if any of the questions that have been added require checking for consistency. If they do, you should add logic to check their consistency in the data-entry program, the editing program, or both. When you add a consistency check, be sure to add a corresponding error message to the data-entry or editing message file. Also, if you add error messages, make sure you do not use an existing error message number.

Some error messages are followed by a *reenter* command that returns to the field that is being entered. This forces the entry operator to address the error before advancing. Because the data-entry operator will at times be required to enter corrections, careful supervision is necessary. When you add your own error messages, consider carefully whether you want to force the data-entry operator to resolve the problem before advancing. If this is the case, follow your error message with a *reenter* command.

ALPHANUMERIC VARIABLES

The data-entry application checks that alphanumeric variables are correctly entered. It performs four checks on each alphanumeric variable. First, it checks that the entered value contains only codes that are listed on the questionnaire (that is, it performs a range check). Second, it checks that the responses are entered in alphabetical order (that is, *ACG* and not *GAC*). Third, it checks that if the 'Doesn't know' or 'No one' codes (generally the letter '*Y'*) are included in the response, then no other response is present (that is, it will not allow the response *ACY*). Fourth, it checks that if the missing code ('*?')* is included in the response, then no other response is present (that is, it will not allow the response *AC?*).

The data-entry application also rearranges the values entered in alphanumeric variables so that each response is stored in the location that defines its sub-item. For the variable 'mn2', for example, the response *ACG* will be rearranged to *A C G* , where there is one blank each between *A* and *C* and *C* and *G* and three blanks after *G*.

USER-DEFINED FUNCTIONS

A nice feature of CSPro is that it allows programs to define their own functions. Such functions are known as user-defined functions and can be useful. In particular, they allow one to avoid rewriting frequently used code. User-defined functions are always defined at the top of a CSPro application. The data-entry application *entry.app* contains 14 user-defined functions. You do not need to modify these functions, but you must understand what they do if you are to understand the data-entry application.

The *valid* function checks whether a variable's value is one of the special values: inconsistent, doesn't know, missing or not applicable. If the value of a variable is not applicable, the *natozero* function changes it to '0', allowing it to be added to another variable (for an example of its use, see procedure *cm9*).

The next seven user-defined functions (*zscoef*, *dabs*, *zspct*, *zseval*, *zscr*, *zsanth* and *agemth*) are used in the Questionnaire for Children Under Five to calculate the anthropometry scores that are found at the end the Anthropometry module. The *agemth* function is called to calculate the child's age in months. The *zsanth* function is then called. This function calls *zseval*, *zscr* and *zspct*. The function *zseval* calls *zscoef*, and *zspct* calls *dabs*. You will only encounter these functions in the anthropometry variables, and if you encounter them you will know that they are calculating and then checking anthropometry scores.

The code in the *agemth* function calculates the age of the child in months. Because anthropometry is highly sensitive to age, the age of the child must be based on the child's age in days. The code first calculates the number of days that have elapsed between the beginning of the year and a child's birth. It then calculates the number of days that elapsed between the beginning of the year and the date of interview. Finally the number of days in the years between the year of birth and the year of interview is added to the number of days since the beginning of the year until the date of interview. The difference between these two numbers of days is the child's age in days. This is then converted into the child's age in months by dividing by 30.4375 (the average number of days in a month over four years). Because of the need for accuracy, the child's age in months is calculated to two decimal places.

The *vdvalid*, *vdoi* and *vdob* functions check that vaccination dates entered in the Immunization module are consistent, are not after the date of interview and are not before the date of birth, respectively. The *endmess* (short for 'end message') function displays a message at the end of a questionnaire that asks the data-entry operator whether he/she wants to review the current questionnaire or continue to the next one. Finally, the *alphachk* function performs the checks on alphanumeric variables detailed in the previous subsection.

DATES AND CENTURY MONTH CODE

The model programs (including the data-entry application) use century month codes (CMC) for most dates. The CMC for a date is the number of months since December 1899. For example, the CMC for January 1900 is 1; the CMC for March 2000 is 1203. The CMC for a date is calculated as follows: subtract 1900 from the date's year, multiply that number of years by 12, and then add the number of the date's month to the product. For example, the CMC for March 2000 is calculated as (2000-1900) x 12 + 3.

The data-entry application uses four functions to simplify working with century month codes. Two of these functions, *setlb* and *setub*, calculate the lower and upper bounds, respectively, for the CMC of the date of an event. The other two functions, *adjlba* and *adjuba*, adjust the lower and upper bounds, respectively, of the CMC of the date of an event (that is, the birthday of a child) when an age is also specified. Table 7.4 below summarizes these functions.

Table 7.4
CSPro Functions for Simplifying Work with Century Month Codes

`lcmc = setlb (month, year, minimum);`	The function's arguments are a *month*, *year* and a *minimum* CMC. If both *year* and *month* are valid, the CMC is calculated and returned. If *year* is not valid, *minimum* is returned. If *month* is not valid, the CMC for January of *year* is returned.
`ucmc = setub (month, year, maximum);`	The function's arguments are a *month*, *year* and a *maximum* CMC. If both *year* and *month* are valid, the CMC is calculated and returned. If *year* is not valid, *maximum* is returned. If *month* is not valid, the CMC for December of *year* is returned.

`t = adjlba (lcmc, ucmc, di, di, age);`
`if t < 0 then`
`  e = errmsg (2171);`
`else`
`  lcmc = t;`
`endif;`
	The function's arguments are a date's minimum CMC (*lcmc*), a date's maximum CMC (*ucmc*), the CMC for the date of interview (*di*), and an *age*. The function raises *lcmc* using *age* and *di*. The function returns the resulting date if it is greater than or equal to *lcmc* and less than or equal to *ucmc*. If the resulting date is less than *lcmc*, the function returns *lcmc*. If the resulting date exceeds *ucmc*, the function returns –1.

`t = adjuba (lcmc, ucmc, di, di, age);`
`if t < 0 then`
`  e = errmsg (2171);`
`else`
`  ucmc = t;`
`endif;`
	The function's arguments are a date's minimum CMC (*lcmc*), a date's maximum CMC (*ucmc*), the CMC for the date of interview (*di*), and an *age*. The function lowers u*cmc* using *age* and *di*. The function returns the resulting date if it is greater than or equal to *lcmc* and less than or equal to *ucmc*. If the resulting date is greater than *ucmc*, the function returns u*cmc*. If the resulting date is less than *lcmc*, the function returns –1.

RECEIPT OF QUESTIONNAIRES FROM THE FIELD

When the questionnaires for a cluster arrive from the field, the questionnaire administrator must check the number of questionnaires against the cluster control sheet. If any questionnaires are missing, the questionnaire administrator must contact the fieldwork team to see if the questionnaires can be found. If they cannot, the fieldwork team must redo the missing questionnaires if at all possible (that is, if the fieldwork team is still located near the cluster). If this is not possible and the missing questionnaire is a Household Questionnaire, the questionnaire administrator must add a blank questionnaire (except for the identification variables) with a result code of '6' to the cluster's questionnaires. If the missing questionnaire is a women's or children's questionnaire, the questionnaire administrator must change the completed questionnaire totals on the cover sheet of the household to which the woman or child belongs.

The arrival of questionnaires from the field must be recorded by the questionnaire administrator on the cluster tracking form. The cluster tracking form is available in both electronic and paper formats. An example of the paper version of the form is found at the end of this chapter; the electronic version of the form is available through the supervisor's menu. Prior to receiving any questionnaires from the field, clusters must be listed in ascending order on the cluster tracking form. As clusters of questionnaires are received, the information for each can then be recorded in the corresponding space. In addition, for each cluster the questionnaire administrator must ensure that:

- All of the questionnaires listed on the cluster control sheet are present
- The household, women's and children's questionnaires for each household are together
- The questionnaires are sorted by household number in ascending order
- All of the women's questionnaires are placed together in ascending order of the women's line number, followed by all of the children's questionnaires, also in ascending order of line number
- All geographic and interview information codes are completed on the Household Information Panel at the start of each Household Questionnaire.

Once the checking is complete and any problems have been resolved, the questionnaire administrator must record the number of household, women's and children's questionnaires on the cluster tracking form. He/she should also create a cluster summary sheet. The cluster summary sheet should be a piece of heavy paper that folds over the end of the package of questionnaires to display the cluster number in bold numbers. The cluster summary sheet should also summarize the number of questionnaires in the cluster, as shown in Table 7.5 below.

Once the cluster summary sheet has been created, the supervisor should assign the cluster to a data-entry operator (note that the assigned data-entry operator must enter all of the cluster's questionnaires). The questionnaire administrator must then record the name of the data-entry operator on the cluster tracking form.

Table 7.5
Sample Cluster Summary Sheet

101

Total Household Questionnaires _____

Total Households Completed _____

Total Women's Questionnaires Completed _____

Total Women Completed _____

Total Children's Questionnaires Completed _____

Total Children Completed _____

Whenever the questionnaires are removed from their storage location for entry or editing, they should always be re-bundled in the same order and returned to the same location. Handling the paper questionnaires in a strict, systematic manner will make the various data-processing steps proceed more quickly and efficiently.

When each cluster has been logged onto the cluster tracking form, the questionnaires should be reviewed by the supervisor and checked that they are ready for data entry. The specific checks that should be performed are listed in Table 7.6 below. Once the checking is complete and any problems detected have been resolved, the supervisor assigns the cluster to a data-entry operator.

MAIN DATA ENTRY

Data entry is best begun soon after interviewers start working in the field. This will allow you to spot and correct mistakes that certain interviewers or teams may be making. Serious problems that may escape the field supervisor's notice can be picked up in time to retrain field staff and correct serious errors. Field supervisors are responsible for checking the questionnaires for completeness and consistency and for classifying any responses the interviewer could not code. Only minimal checking as described above should be necessary once the questionnaires are returned to the office. As soon as data from one cluster arrive back at headquarters, data entry should begin.

Table 7.6
Office Checks

1. Ensure that all Household Questionnaires are sorted in ascending order of household number within the cluster.
2. Ensure that all geographic and interview information codes are completed on the Household Information Panel at the start of each Household Questionnaire.
3. The eligibility for interview of each member of the household should be checked on the Household Listing Form. To be eligible for the women's modules, a person must be female and aged 15 through 49. To be eligible for the Child Labour module, a person must be aged 5 through 14 (or the country-specific age range). To be eligible for the Child Health modules, a person must be under age five. For each eligible person, the eligibility code should have been circled for their line number or the line number of their mother or primary caretaker (HL6-HL8, Household Listing Form). For other members of the household, the eligibility columns should be blank on the Household Listing, and will be entered as 00. Follow the instructions in the editing guidelines (Appendix Six) for messages 0101-0132 to resolve any problems.
4. The total number of eligible women and children recorded on the Household Information Panel must match the number of eligible women and children recorded on the Household Listing Form.
5. The total number of eligible women and children recorded on the Household Information Panel must match the number of women's and children's questionnaires for the household.
6. The number of complete women's and children's interviews recorded on the Household Information Panel must match the number of women's and children's questionnaires with result code '1' ('Complete') for the household.
7. Ensure that all of the Questionnaires for Individual Women are placed together in ascending order of the women's line number (WM4), followed by all of the Questionnaires for Children Under Five in ascending order of the child's line number (UF4).

The process of entering data is guided by a CSPro application named *entry.ent.* This application performs two main tasks. First, for a given household, it ensures that all questionnaires (and only these questionnaires) are entered and that all the information on the questionnaires (and only this information) is entered. In particular, the data-entry application begins by entering all of the data from a Household Questionnaire. Once these data have been entered, it reviews the Household Listing and determines which household members are eligible for women's questionnaires. Having identified these members, the data-entry application asks for each individual's data from

her women's questionnaire, in ascending order of line number. The data-entry application then executes the same procedure for household members who are eligible for the children's questionnaire. When all of a household's questionnaires have been entered, the data for the entire household is saved and the application moves on to the next household in the cluster, if any.

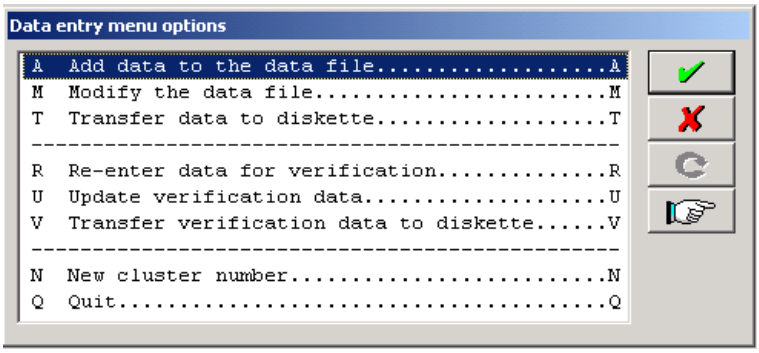
Because data are saved only after the questionnaires for the household and all eligible women and children have been entered, data-entry operators should not leave their computers in the middle of entering data for a household. Before taking a break or stopping work for the day, all of the questionnaires for a household must be completely entered. Further, it is recommended that data be copied onto the supervisor's computer or a floppy diskette as a precautionary measure. In addition, every evening the supervisor must copy the contents of *c:\mics* and all of its subdirectories onto the secondary storage device. This safeguard will allow the supervisor to recover the data if her/his computer crashes.

In addition to controlling which questionnaires are entered, the data-entry application rigorously controls the skip pattern within a questionnaire. That is, it will only ask for the responses to questions that should have been asked given the responses to previous questions. For example, if the value 2 is entered in variable *cm1* (that is, the woman has never given birth), the data-entry application will next ask for a value for variable *ma1*, skipping all variables that pertain only to women who have given birth.

The second task of the data-entry application is to minimize data-entry errors. The data-entry application does this by performing checks as the data are entered. If a value entered for a question is outside the range of values on the questionnaire or if some other basic inconsistency is detected, the data-entry application displays an error message and requires the data-entry operator to resolve the inconsistency before advancing. More complex inconsistencies, whose resolution would slow data entry considerably, are not checked during the data-entry process and are checked instead during secondary editing.

THE DATA-ENTRY MENU

The data-entry menu simplifies the task of entering the data. The data-entry menu is created by the CSPro application *entry_menu.bch* (you do not need to modify this application). The illustration below shows the data-entry menu.



Options A, M and T are used only if the data-entry operator is entering the main data file. Options R, U and V are used only if the data-entry operator is entering the verification data file. Options N and Q can be used under either circumstance.

Option A allows the data-entry operator to add cases to the main data file, option M allows her/him to modify existing cases in the main data file, and option T allows her/him to copy the completed main data file to a diskette so that it can be transferred to the supervisor's computer. Option R allows the data-entry operator to add cases to the verification data file, option U allows her/him to modify existing cases in the verification data file, and option V allows her/him to copy the completed verification data file to a diskette so that it can be transferred to the supervisor's computer.

Option N allows the data-entry operator to change the cluster number so that she/he can enter data for another cluster. Option Q exits the data-entry menu.

DIRECTORY STRUCTURE ON A DATA-ENTRY COMPUTER

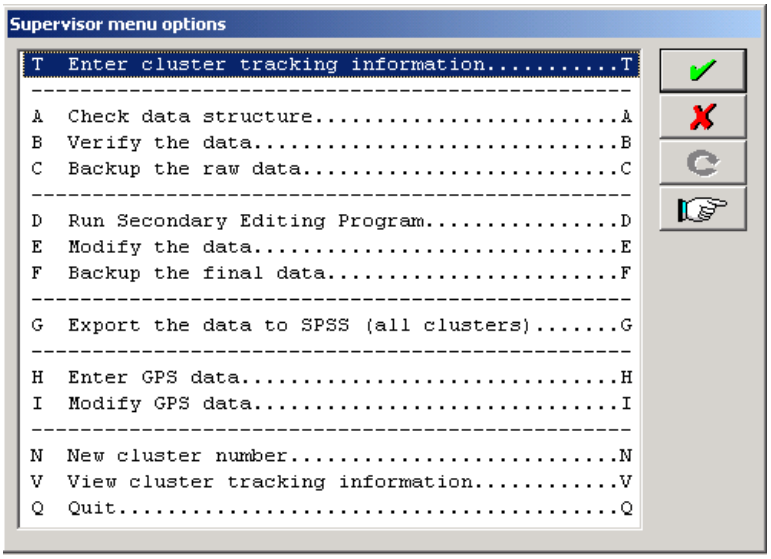
On a data-entry computer, all files and programs are listed in the directory *c:\mics\CSPro* or one of its subdirectories. The subdirectories are named *data*, *dicts*, *entry* and *veri*. The *data* directory contains any main data files that have been entered on the computer. The *dicts* directory contains all the data dictionaries. The *entry* directory contains the data-entry application and the application that creates the data-entry menu. The *veri* directory contains any verification data files that have been entered on the computer.

Data-entry operator's computer directory structure:

```
MICS
        CSPRO
                DATA            Main data directory
                DICTS           Data dictionaries directory
                ENTRY           Data-entry application directory
                VERI            Verification data directory
```

THE SUPERVISOR'S MENU

The supervisor's menu helps the data-processing supervisor to manage the MICS3 data-processing system. It is launched by executing the CSPro application *super_menu.pff.* The menu first asks for the cluster number to process. Once the cluster number is entered, the illustration below shows the supervisor's menu and is followed by a summary of each option.



OPTION T: ENTER CLUSTER TRACKING INFORMATION

This option displays the electronic cluster tracking form so that the data-processing supervisor can enter the cluster tracking information. Information should be entered when the questionnaires for a cluster are first received from the field, when the cluster is assigned to a data-entry operator for main data entry, and when the cluster is assigned to a data-entry operator

for verification data entry. The other information in the electronic cluster tracking form is updated automatically as the supervisor progresses through the supervisor's menu.

OPTION A: CHECK DATA STRUCTURE

This option checks the structure of the cluster's data file by executing the application *check.bch*. After checking the structure of the data file, *check.bch* produces an output file (that is displayed automatically on the screen) that summarizes the number of each type of questionnaire and shows how many of the questionnaires were complete. The data-processing supervisor must check this information against the cluster tracking form and ensure that the two sources agree. If they do not, the data-processing supervisor must identify the problem (for example, the data-entry operator forgot to enter a household) and resolve it by carefully checking the cluster's questionnaires. Once the problem has been resolved (whether by updating the data file or updating the cluster tracking form), the data-processing supervisor must rerun *check.bch*. Only when *check.bch* produces the same number of questionnaires as the cluster tracking form may the data-processing supervisor assign the questionnaires to a second data-entry operator for verification data entry.

OPTION B: VERIFY THE DATA

This option compares a cluster's main data file to its verification data file using CSPro's comparison tool. If there are any differences between the data files they will be displayed on the screen. This output must be printed and given to the data-entry operators responsible for entering the cluster. Working in tandem, the data-entry operators must consult the questionnaires and determine the correct value for each instance in which their data files disagree. Once they have determined the correct values, each operator must update her/his data file. At this point the files must be compared again. When no differences between the two files remain, processing of the cluster can proceed.

OPTION C: BACK UP THE RAW DATA

This option backs up the raw data by copying the verified main data file to the *back-up* subdirectory on the supervisor's computer. It should be run after the structure checks and verification are complete and before any secondary editing is done. The raw data are backed up to document the state of the data before they were edited.

OPTION D: RUN SECONDARY EDITING PROGRAM

This option checks for complex inconsistencies by executing the *editing.bch* application. The output from this program is displayed on the screen and should be printed if it includes any error messages. If this is the case, the list of inconsistencies and the questionnaires for the cluster should be given to a secondary editor. The secondary editor, drawing upon her/his knowledge of the questionnaire and the editing manual in Appendix Seven, will resolve each of the listed

inconsistencies. When the secondary editors have finished their work, they return the list of inconsistencies and required actions to the data-processing supervisor, who implements the changes that they required (see Option E below). The data-processing supervisor then reruns *editing.bch*. If there are no error messages, processing of the cluster can proceed; if there are error messages, a list of them and the questionnaires must be given to the secondary editor for further editing. In some cases, messages will be considered acceptable by the secondary editor and there is no further need to correct the data. If the only messages that remain are those the secondary editor considers acceptable, then the process of secondary editing is complete.

OPTION E. MODIFY THE DATA

This option executes the *entry.ent* application so that the data-processing supervisor can implement the changes required by the secondary editors. After using this option, the data-processing supervisor should return to Option D to make certain that changes to the data have corrected the inconsistencies as desired and that no new inconsistencies have been created.

OPTION F. BACK UP THE FINAL DATA

This option copies the final data files to the *final* subdirectory on the data-processing supervisor's computer. The data stored in this directory will later be concatenated and then exported to SPSS.

OPTION G. EXPORT THE DATA TO SPSS

This option concatenates all of the data files in the *final* subdirectory into one file and then exports this data file by executing *export.bch*. The application produces four ASCII text files and a corresponding SPSS description file for each text file. One text file contains households, one contains household members, one contains women and one contains children.

OPTION H. ENTER GEOGRAPHIC POSITIONING SYSTEM (GPS) DATA

This option allows the data-processing supervisor to enter GPS location data by executing the *gpsentry.ent* application. Unlike the main data-entry program, this application allows the data-processing supervisor to enter as many clusters at a time as he/she would like. The application requires the data-processing supervisor to enter the GPS data twice as a check against keying errors.

OPTION I. MODIFY GPS DATA

This option allows the data-processing supervisor to modifying GPS location data by executing the *gpsentry.ent* application.

OPTION V. VIEW CLUSTER TRACKING INFORMATION

This option displays the information stored in the cluster tracking form for all the clusters.

OPTION N. SELECT NEW CLUSTER

This option changes the cluster number so that the data-processing supervisor can begin processing a new cluster.

DIRECTORY STRUCTURE ON THE DATA-PROCESSING SUPERVISOR'S COMPUTER

The MICS3 data-processing system uses a particular directory structure on both data-entry computers and the data-processing supervisor's computer. The structure of data-entry computers is discussed below. On the supervisor's computer, all files and programs related to SPSS are stored in the directory *c:\mics\spss* or one of its subdirectories. On the supervisor's computer, all files and programs related to CSPro are stored in the directory *c:\mics\CSPro* or one of its subdirectories. The subdirectories are named *backup*, *dicts*, *entry*, *export*, *gps*, *final*, *raw*, *super* and *weights*.

Supervisor's computer directory structure:

```
MICS
        CSPRO
                BACKUP          Back-up directory
                DICTS           Data dictionaries directory
                ENTRY           Data-entry application directory
                EXPORT          Export application directory
                FINAL           Final edited data directory
                GPS             GPS data-entry application directory
                RAW             Raw data directory
                SUPER           Supervisor editing application directory
                WEIGHTS         Sample weights application directory
```

The *back-up* directory contains a back-up of data files that have been structurally checked and verified, but not edited. The *dicts* directory contains all the data dictionaries. The *entry* directory contains the data-entry application and the application that creates the data-entry menu. The *export* directory contains the programs used to export the data from CSPro to SPSS. The *final* directory contains a back-up of data files that have been structurally checked, verified and edited. The *raw* directory contains the data files that have been transferred from the data-entry machines. The *super* directory contains the applications that perform structural checks, verification and secondary editing and the application that creates the supervisor's menu. The *weights* directory contains the spreadsheet that calculates sample weights.

STRUCTURE CHECKING

It is essential that the data be structurally sound. The data-entry program enforces most structural coherence, but it cannot check everything without being seriously constrained. It is therefore necessary to execute a structure checking program after main data entry is complete. The structure checking program makes sure that the number of questionnaires in the data file matches the number of questionnaires that arrived from the field and performs a few additional checks on the structure of an individual questionnaire.

The structure checking program is named *check.bch*. This section will focus on what the program does rather than on how it does it. The program is complex and does not lend itself to facile explanations. The best way to understand the logic in the program is to study it carefully once you understand what the program is doing. A sample of the critical output from *check.bch* is shown below.

```
MICS Data Structure Check
Cluster:  3

  Households  |     Women      |    Children
Total Comp Incomp | Eligible  Interviewed | Eligible  Interviewed
         | HH12 FOUND  HH13 FOUND | HH14 FOUND  HH15 FOUND
  2  1    1 |  5   5   4   4 |  4   4   3   3
```

The first block of the output is a summary of the total number of households and their response codes. The second block of output presents the results of four counts of the number of eligible women. The counts in the *hh12* and *hh13* columns are the number of eligible and interviewed women according to the Household Information Panel. The counts in the *found* columns are the number of women's questionnaires and completed women's questionnaires, respectively, in the data file.

The third block of output presents the results of four counts of the number of eligible children under the age of five. The counts in the *hh14* and *hh15* columns are the number of eligible and interviewed children according to the Household Information Panel. The counts in the *found* columns are the number of children's questionnaires and completed children's questionnaires, respectively, in the data file.

The output of *check.bch* must be printed by the supervisor and the information that it contains compared to the cluster tracking form. If there is a difference between the two counts of the questionnaires, the supervisor and the data-entry operator must then use the error listing and the cluster's questionnaires to determine what caused the structural problem. When these causes have been identified, they must be corrected by the data-entry operator. The structure checking application must then be rerun to check if the problem has been fixed without introducing a new problem. Only when the counts generated by *check.bch* match those on the cluster tracking form can verification data entry begin.

The application *check.bch* also produces a list of all of the households in the cluster. Each household's number and result code are displayed, along with a count of women's and eligible children's questionnaires if the household interview was completed (that is, the household result code is equal to 1). A sample of the output for one household is shown below.

```
MICS Data Structure Check
Household:  1
Result:   1

    Women       |    Children
 Eligible  Interviewed | Eligible  Interviewed
HH12 FOUND  HH13 FOUND | HH14 FOUND  HH15 FOUND
  4   4   3   3 |  2   2   1   1
```

The listing of households can be useful in identifying the source of a problem at the cluster level. Suppose, for example, that the cluster tracking sheet lists 20 households in the cluster but only 19 are found in the data file. By comparing the listing of households to the cluster's questionnaires, you can identify which household was not entered.

VERIFICATION

Verification of double-entered data is done by a CSPro comparison application. The comparison application is named *compare.cmp*. It contains a list of all variables (items). As the program is currently configured, checked items will be compared during verification and unchecked items will not be compared during verification. Only one variable is unchecked (the data-entry operator's code, *hh16*) and it is recommended that *no additional variables* be unchecked since differences in other variables can affect the quality of the data.

The comparison application compares the main data-entry file (which has been copied onto the supervisor's machine) to the verification data-entry file (which has been copied onto a diskette, or network drive, if using a network) and produces a list of differences, if any. If there are no differences, the supervisor should back up the raw data and then proceed to secondary editing.

If there are differences, the list of them should be printed and given to the two data-entry operators. The data-entry operators then use the list of differences and the cluster's questionnaires to check each difference and record on the list which file needs to be corrected. When all of the differences have been investigated, the data-entry operators correct any errors in their files. They then recopy the data files to their floppies (or to the network) and the files are compared again. This process continues until the files are identical.

SECONDARY EDITING

Experience has shown that simple inconsistencies can be usefully identified and corrected during data entry. More complex consistency errors, however, must be resolved by carefully examining the questionnaire. This type of consistency checking is best carried out as a separate step, with

errors reported on a printout that can be used for marking the corrections. This step is known as secondary editing.

The secondary editing program is named *editing.bch*. It performs a long list of consistency checks (for example, are ages and dates of birth consistent?) and outputs a list of the inconsistencies found in the raw data file. The data-processing supervisor must print out this list and give it and the cluster's questionnaires to one of the secondary editors. The secondary editor then reviews the list of errors and the responses on the questionnaire. Using the editing guidelines (found in Appendix Seven) and her/his knowledge of the questionnaire, the secondary editor then either writes a correction on the error listing or writes that no action is to be taken. When the secondary editor has reviewed each and every error message, he/she returns the annotated error listing to the data-processing supervisor. The data-processing supervisor then makes the suggested changes in the raw data. When this has been done, the data-processing supervisor reruns the editing program. Only when the editing program produces no error messages may processing of the cluster continue.

Three aspects of the editing process are vitally important. First, for every error the secondary editor must carefully examine the questionnaire concerned. Second, the secondary editor must always refer to the editing guidelines before developing a solution to the problem. Third, the editing process must be repeated until there are no errors remaining.

Once this third task has been completed, the data-processing supervisor can back up the edited data. These data are now considered to be clean and can be used to construct analysis files.

EXPORTING THE DATA TO SPSS

When primary data processing is complete, you will have a clean data file for each cluster in your sample. While primary data processing is done using CSPro, secondary data processing is done primarily in SPSS. The first step in secondary data processing is therefore converting the data from CSPro's data format to SPSS' data format. This is done using the 'Export the data to SPSS' option on the supervisor's menu.

When you select this option, all of the data files in the *final* subdirectory (that is, all of the data files that have been verified, checked and edited) are concatenated into a single data file named *all.dat*. This data file is then exported to SPSS by the *export.bch* application. This application creates four ASCII data files (*mych.dat*, *myhh.dat*, *myhl.dat* and *mywm.dat*) and four SPSS description files (*mych.sps*, *myhh.sps*, *myhl.sps* and *mywm.sps*) in the directory *c:\mics\spss*. While the SPSS data description files will read the ASCII data files into SPSS, they will not save them. To get the data description files to save the data in SPSS format, the SPSS command

```
save outfile = `filename.sav'.
```

must be added to the end of each data description file. The word *filename* should be replaced by *hh*, *hl*, *wm* or *ch*, depending upon the type of data file. Once this command has been suitably

modified and added to each data description file, executing the SPSS data description files will create the SPSS data files *hh.sav*, *hl.sav*, *wm.sav* and *ch.sav*.

CREATING AN ANALYSIS FILE

Formatted: Highlight

The structure of the data file during primary data processing simplifies the process of entering the data. This structure is not optimal for analysing the collected data, however, so the first task after the data have been transferred to SPSS is recoding variables to make analysis easier and more efficient. This task is known as creating an analysis file. This section will detail the steps involved in creating MICS3 analysis files. The analysis files that will result from following these steps can be used by the model tabulation plans and are suitable for distribution to researchers.

RECODING VARIABLES

The SPSS programs *makehl.sps*, *makewm.sps* and *makech.sps* recode existing variables to create new ones. Variables that are used in several tabulations are recoded in these programs and then saved; all other recoding is done in the tabulation programs and is temporary.

The recoding of most variables uses standard SPSS commands and will not be discussed here. There is, however, one frequently used approach that must be explained: the recoding of variables into 0 or 100. This unusual recoding is done for presentation purposes only. When SPSS displays percentages in a table, it displays all of a variable's categories. For many tables in the tabulation plan, we are only interested in one category. If we assign a value of 100 to that category and a value of 0 to all other categories, the mean of the variable is the percentage of respondents in that category. Thus, asking SPSS to display the mean of the new variable will result in only the percentage we are interested in being displayed.

For example, the variable *ha1* records whether a woman has heard of AIDS. It takes a value of 1 if a woman has heard of AIDS and a value of 2 if she has not. We are interested in displaying the percentage of women who have heard of AIDS. In the program *makewm.sps*, the variable *ha1* is recoded into the variable *aids*. The variables *aids* takes a value of 100 if the woman has heard of AIDS and a value of 0 otherwise. The mean of the variable *aids* is the percentage of women who have heard of AIDS. To understand why this is so, consider the example below:

Women who have heard of AIDS	10
Total number of women	20
Percentage of women who have heard of AIDS	10 / 20 x 100 = 50
Mean of the variable AIDS	(10 x 100 + 10 x 0)/20 = 10 x 100/20 = **10/20 x 100 = 50**

CALCULATING AND ADDING SAMPLING WEIGHTS

If separate sampling frames were used for different regions (or domains) at the first stage of sampling, the national sample was not chosen with probability proportional to size. This may also happen if you stratified according to some other factor (for example, urban/rural or slum/non-slum) and took different sampling fractions (proportions) in each stratum. These samples are not self-weighting, and you must weight your sample when you report national estimates. That is, you must ensure that each separate sub-sample – for example, each separate region (or domain) – contributes only what it would contribute if the survey sample at the national level had been chosen with probability proportional to size.

If your sample is not self-weighting, you must calculate sample weights and add them to your analysis files. This task is accomplished by using the spreadsheet *weights.xls* and the SPSS programs *weights_table.sps*, *weights.sps* and *weights_merge.sps*. The spreadsheet is used to calculate the sample weights. It has two worksheets, *calculations* and *output*. The *calculations* worksheet performs the calculations. The *output* worksheet contains only the sample weights and a list of cluster numbers; this format is useful for reading the data into SPSS. The program *weights_table.sps* produces data needed for calculating the sample weights. The program *weights_merge.sps* adds the appropriate sample weights to the analysis files. The program *weights.sps*, which you will never directly execute, describes the structure of the data in the output worksheet.

The process of calculating sample weights and adding them to your analysis files can be broken down into seven steps:

Step 1: Adjust the number of rows in the *calculations* and *output* worksheets so that there is one row per cluster in your survey. After you have added or deleted rows, be sure to check that doing so did not affect the totals row in the *calculations* worksheet.

Step 2: Enter the weights that were built into the design of the sample into *weights.xls*. If your weights vary across clusters within a particular stratum or domain, you must complete both the cluster sampling fraction column and the stratum (or domain) sampling fraction column with the information provided by your survey's sampling expert. If your weights vary across strata (or domains), but not across clusters within strata (that is, the sample is self-weighted within strata or domains), enter the value 1 in the cluster sampling fraction column and complete the stratum (or domain) sampling fraction column using the information provided by your survey's sampling expert.

Step 3: Update the definition of strata (or domains) on lines 3 through 10 of the program *weights_table.sps*. The standard programs assume that strata are formed by all combinations of area (that is, urban and rural) and region and that there are four regions (the program should be modified to reflect the strata or domains in use in your sample).

Step 4: Execute the program weights_table.sps.

Step 5: Copy the information in the table and paste it into the *calculations* worksheet of *weights.xls*. When you complete this step, *weights.xls* will automatically calculate the sample weights.

Step 6: Save the *output* worksheet as a comma-separated value file *(*.csv)* under the name *weights.csv* in the directory *c:\mics\weights*.

Step 7: Execute the program *weights_merge.sps*. Once you have completed the seventh step, be sure to check the output list for error messages and to open the analysis files and confirm that the weights have been properly merged.

CALCULATING AND ADDING A WEALTH INDEX

The MICS3 tabulation plan includes as a background variable a household wealth index. This wealth index is calculated by the program *wealth.sps*, which creates a data file *wealth.sav* that contains identification variables, a variable containing each household wealth score and a variable containing each household's wealth index. The program *wealth.sps* first produces frequencies of all household variables concerned with wealth or assets. It then recodes variables describing household and individual assets into dichotomous variables. The program then uses factor analysis (specifically principal components analysis) to calculate a wealth score for each household. Finally, it uses the wealth score to create household wealth quintiles (that is, the wealth index) and then saves them in an SPSS data file. The choice of variables to be included in the factor analysis is critical, and should not be made without carefully consulting the frequencies produced. Information on variables included in the analysis is available at www.childinfo.org. Once the wealth index has been calculated, executing the program *wealth_merge.sps* will add it to your analysis files. Be sure to check the output list for error messages and to open the analysis files and confirm that the wealth index has been properly merged.

ADDING GPS READINGS

Some MICS3 surveys will take Geographic Positioning Systems (GPS) reading for their clusters during fieldwork. GPS readings, which precisely locate clusters, can be used to add other geographical data sets (for example, rainfall data) to a MICS3 data set. If your survey is taking GPS readings, you will want to merge them onto your analysis files. This task is accomplished by two SPSS programs, *gps.sps* and *gps_merge.sps*. The program *gps.sps*, which you will never directly execute, describes the structure of the data file *gps.dat* (which is created by the CSPro data-entry application *gpsentry.bch*). If you have changed the CSPro dictionary *gps.dic*, you must update *gps.sps* to reflect your changes. The program *gps_merge.sps* merges GPS readings onto the analysis files. You should not need to modify this program. To merge the GPS readings

onto the analysis files, execute *gps_merge.sps*. Be sure to check the output list for error messages and to open the analysis files and confirm that the GPS readings have been properly merged.

TABULATION

Chapter 8 describes in detail the process of analysing the data and preparing reports. Running tabulations is a major component of this activity. A model tabulation plan (Chapter 8 and Appendix Six) and tabulation programs for SPSS accompany this manual. There is one SPSS tabulation program for each table in the MICS3 tabulation plan. Each program's name is the letter 't' followed by the number of the table in the tabulation plan. For example, the program *t1.sps* creates Table 1 in the tabulation plan.

Prior to running tabulations for a report, it is essential to produce a set of (unweighted) frequency distributions for every variable in the data file. The frequencies should be checked for unusual values, those that are outside the range of most responses, and those that seem implausible answers to the relevant question. For example, a response of '53' to the question on the number of hours a child did chores in the last week seems both implausibly high and too precise. The identification information for such cases should be written out and the values in the data file checked against the original questionnaires.

REVIEWING THE MODEL PROGRAMS

Each tabulation program must be carefully reviewed. It is important to check whether the variables used in the tabulation program exist in your data file. If they do not, check whether the variable is of primary or secondary importance. If a variable of primary importance does not exist in your data file, you must either remove the table entirely or ask an analyst to redesign the table. If a variable of secondary importance is missing, remove all references to the variable in the tabulation program and make any other adjustments necessitated by its absence.

All recoding activity must also be carefully checked. If there are variables on your questionnaire that have non-standard categories, any recoding activity involving those variables must be examined. If your questionnaire contains non-standard variables, they must be recoded if they are to be used in any tabulation.

You must also check any merge operations if your questionnaire uses case identifiers not present in the standard questionnaire. There are a number of merges in the tabulation programs that will only work if unique identifiers are used.

APPLYING THE SAMPLE WEIGHTS

Weighting in the tabulation programs is straightforward except where the SPSS *aggregate* command is involved. If the goal of the *aggregate* command is to cumulate across cases to calculate a numerator and a denominator, weights must be applied before the *aggregate*

command. They should not be used when working with the resulting file; it has already been weighted.

For example, Table HH.1, shown in Appendix Seven, contains the household responses rate. The household response rate is difficult to calculate because it requires dividing one variable by another within the table. One solution to this problem is to create an aggregate file that contains counts of sampled households, occupied households and interviewed households. The aggregate file will contain one case for each category of the specified break variable (for example, urban/rural).

The weights must be applied when the aggregate file is created to generate the weighted numerator (the count of interviewed households) and the weighted denominator (the count of occupied households). Once the aggregate file has been created, the household response rate for each category of the break variable is the numerator divided by the denominator.

If the goal of the *aggregate* command is to create a summary statistic for individual cases, weights must be applied after the *aggregate* command. For example, Table HH.3 in Appendix Seven contains information on the percentage of households that contain at least one child under the age of 15.

This information is not present in the household data file, but it can be created by aggregating the household listing file. The break variables are cluster number and household number. Weights are applied after aggregating because we are interested in the weighted percentage of households with at least one child under the age of 15, not the weighted number of children under the age of 15 in each household.

THE INCLUDE COMMAND

The SPSS program *tables.sps* will run all of the tabulation programs at once. It consists of a series of SPSS *include* commands that execute the tabulation programs individually. If SPSS encounters an error in a program that is included (that is, executed by an *include* command), it will immediately stop executing the program and return to the program that included the program (that is, the program that contained the *include* command).

Because of this, you should only use *tables.sps* when you have checked, modified and tested all of the individual tabulation programs. Be sure to also remove any *include* command that executes a tabulation program that you are not using.

The *include* command imposes four restrictions on the programs that it executes. The first restriction is that each command must begin in the first column of the program. This restriction appears to limit program indentation, but a line may be indented if it begins with the '+' character. The commands below illustrate the use of the '+' character.

```
do if (cage >= 6 and cage <= 9).
+ compute solids = 0.
+ if (BF3G = 1) solids = 100.
end if.
variable labels solids "Solid foods".
```

The second restriction imposed by the *include* command is that if a command continues over multiple lines, column 1 of the continuation lines must be blank. The example below illustrates a multi-line command that respects this restriction.

```
add files
 /file=*
 /file='tmp6.sav'.
```

Notice that the subcommands on the second and third lines are indented two columns. (While they need only be indented one column to satisfy the restriction, they have been indented two columns to remain consistent with the MICS3 programming style.)

The third and fourth restrictions imposed by the *include* command are that command terminators are optional and that an asterisk (*) in the first column of a line indicates a comment line. Neither of these restrictions affects our tabulation programs.

In addition to *tables.sps*, there is an SPSS program that automates the creation of analysis files. This program is named *CSPro.sps*. This program should only be used when all of the component programs have been executed and shown to work. It is useful for recreating analysis files when a change is made to one of the file creation programs. It ensures that all of the analysis file creation programs will be executed in the proper order.

ARCHIVING AND DISTRIBUTING DATA

An important – but often neglected – component of data processing is the archiving and documentation of data files. In addition, whether the data files will be available widely or only within a single institution, it is imperative to establish some guidelines for distribution well in advance. These steps – archiving, documenting and distributing – require an investment of time and effort. The investment is well worth it, however, for a number of reasons:

- **Increasing the cost-effectiveness of data collection.** Collecting survey data is a costly and labour-intensive activity. In order to justify this investment, the data collected should be exploited as fully as possible. Making data files available to other researchers increases the cost-effectiveness of the survey activity.

- **Increasing country ownership of the data and acceptance of the results.** When the data file is available for others to use, the data collection process gains credibility. The collectors of the data are viewed as having confidence in their findings, and the

accessibility of the data file to other researchers means that the results can be replicated and verified by others.

- **Ability to examine trends.** Often, published results from different surveys are not directly comparable. For example, one survey report may define adult respondents as those age 15 or older while another defines adults as those age 18 or older. Without data files, the best that can be done is an imprecise comparison of the two sets of results. When the data files for the two surveys are available, however, the results can often be re-tabulated so that they are directly comparable, allowing conclusions about trends to be drawn.

- **Ability to compare results within and across countries.** It is often instructive to compare results across countries, either within a subregion or across regions. These comparisons facilitate the identification of areas where a particular programme emphasis is needed or where programmes have been particularly successful. Furthermore, it may be useful to compare results from different surveys within the same country. Sometimes this is done to validate unexpected results (when infant mortality is lower than expected, for example) or to assess the effects of a particular data collection methodology (for example, relying on vaccination cards versus mothers' reports of vaccinations). In order to conduct these types of analyses, researchers require access to data files so that directly comparable figures can be calculated.

- **Allows in-depth analysis of important subject areas by specialists.** Because of the pressure to report findings quickly, the information presented in a survey report usually includes only the basic findings of a survey. A well-documented and available data file will allow in-depth analyses of particular subject areas to be conducted, and these analyses can be done by subject specialists who may not be on the staff of the data collection institution.

The MICS3 analysis file should be archived, documented and distributed. At a minimum, the documentation accompanying the analysis files should include a copy of the full report, a copy of the questionnaire and a description of the sample design. The documentation should also include a codebook containing the location and description of each variable in the analysis file (this can be easily created in SPSS). Copies of all of the programs and files used during the survey processing should also be archived and made available upon request. A copy of the analysis files and their documentation should be sent to the UNICEF Regional Office and to UNICEF New York (Statistics and Monitoring Section). Finally, a policy and procedure for the distribution of the data file to others should be established.

Table 7.7
Checklists

Before fieldwork:

- Obtain computers and other data-processing equipment.
- Set up a data-processing room or space.
- Recruit a data-processing supervisor and other personnel.
- Set up a system for organizing processing activities.
- Adapt programs for consistency with pre-test questionnaire.
- Enter and edit pre-test questionnaires.
- Finalize programs based on pre-test experience and the final questionnaire.

During fieldwork:

- Receive questionnaires from the field.
- Assign main data entry.
- Check the structure of the main data-entry file.
- Assign verification data entry.
- Verify that the main and verification data files are identical.
- Back up the raw data file.
- Perform secondary editing.
- Back up the final data file.

After fieldwork:

- Export the data to SPSS.
- Recode variables.
- Calculate and add sample weights, a wealth index and GPS data.
- Run the tabulation programs.
- Archive the data and develop a data distribution policy and system (for example, a website).
- Send the analysis files, their documentation and all programs to UNICEF.

Table 7.8
Sample Cluster Tracking Form

Cluster number	Date received	Households selected	Number of questionnaires			Data entry			Date check complete	Date verification complete	Date of raw data back-up	Date of editing	Date of final back-up
			Household	Women's	Children's	Operator name	Operator number	Creation date					
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							
						M:							
						V:							