END-DECADE
MULTIPLE-INDICATOR DATA PROCESSING MANUAL

EPIINFO

00.4.20

Overview

Data processing personnel

The data processing team for a MICS survey includes four types of personnel: questionnaire administrators, 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.

Questionnaire administrators

The questionnaire administrators check the clusters that arrive from the field. They check that all of the questionnaires are present and ready to be entered. If there are missing questionnaires, they must contact the fieldwork team and try to find them. If the missing questionnaires cannot be found, the questionnaire administrators must resolve the problem.

In the case of household questionnaires, this means requiring the fieldwork team to redo the household interview. If this is impossible, the questionnaire administrators must add a blank household questionnaire with the result code 5.

In the case of women's and children's questionnaires, this means requiring the fieldwork team to redo the missing questionnaires. If this is impossible, the questionnaire administrators must change the completed questionnaire totals on the household cover sheet.

Data entry operators

The data entry operators enter the data. They should have prior data entry experience and be familiar with the questionnaires. Before beginning data entry in earnest, a two or three day training must be held. By the end of the training, the data entry operators should be comfortable with the data entry program and aware of their responsibilities.

Secondary editors

Secondary editors investigate complex inconsistencies discovered during the data entry process. They must have an excellent grasp of the questionnaires and the goals of the survey. Editing guidelines are provided to aid them during the secondary editing process.

Data processing supervisor

The data entry supervisor oversees the data processing system. The data entry supervisor must have an excellent grasp of the questionnaire and programming skills in the data entry package and SPSS. The data entry supervisor also resolves problems encountered by the data entry operators.

Data processing equipment

The list below shows the equipment that is necessary for data processing.

- Data entry machines
- A supervisor's machine
- A secondary storage device (e.g., a Zip or Jaz drive)

- Diskettes
- A printer
- Paper
- Toner cartridges/printer ribbons
- Surge protectors
- Uninterrupted power supplies
- Green pens
- A data entry room
- An editing room

The data entry machines should have at least 486 processors, Windows 95 or higher, 16 megabytes of RAM or more, 500 megabytes of hard disk space or more and a 3.5’’ floppy diskette drive. The supervisor’s machine should have a 300 Mhz processor, Windows 95 or higher, 64 megabytes of RAM or more, 500 megabytes of hard disk space or more, a 3.5’’ floppy diskette drive and a CD-ROM drive. The secondary storage device should be connected to the supervisor’s machine.

The Uninterrupted power supplies are essential if the country in which you are working suffers from power outages. The green pens are for the data entry clerks and the office and secondary editors. They should be used whenever a questionnaire is modified. The green ink distinguishes their changes from the original value recorded by the interviewer and the changes made by the fieldwork team.

The data entry room should be large and cool. Each data entry agent should have space for their machine and the questionnaire they are working on. The editing room is for the office and secondary editors. It must contain shelves or cupboards in which the questionnaires can be stored in an organized fashion.

Data processing design

MICS data processing is organized around clusters. The data for each cluster are entered into a set of data files whose names include the cluster’s number. This approach breaks data processing up into discrete segments and keeps the size of data files to a minimum.

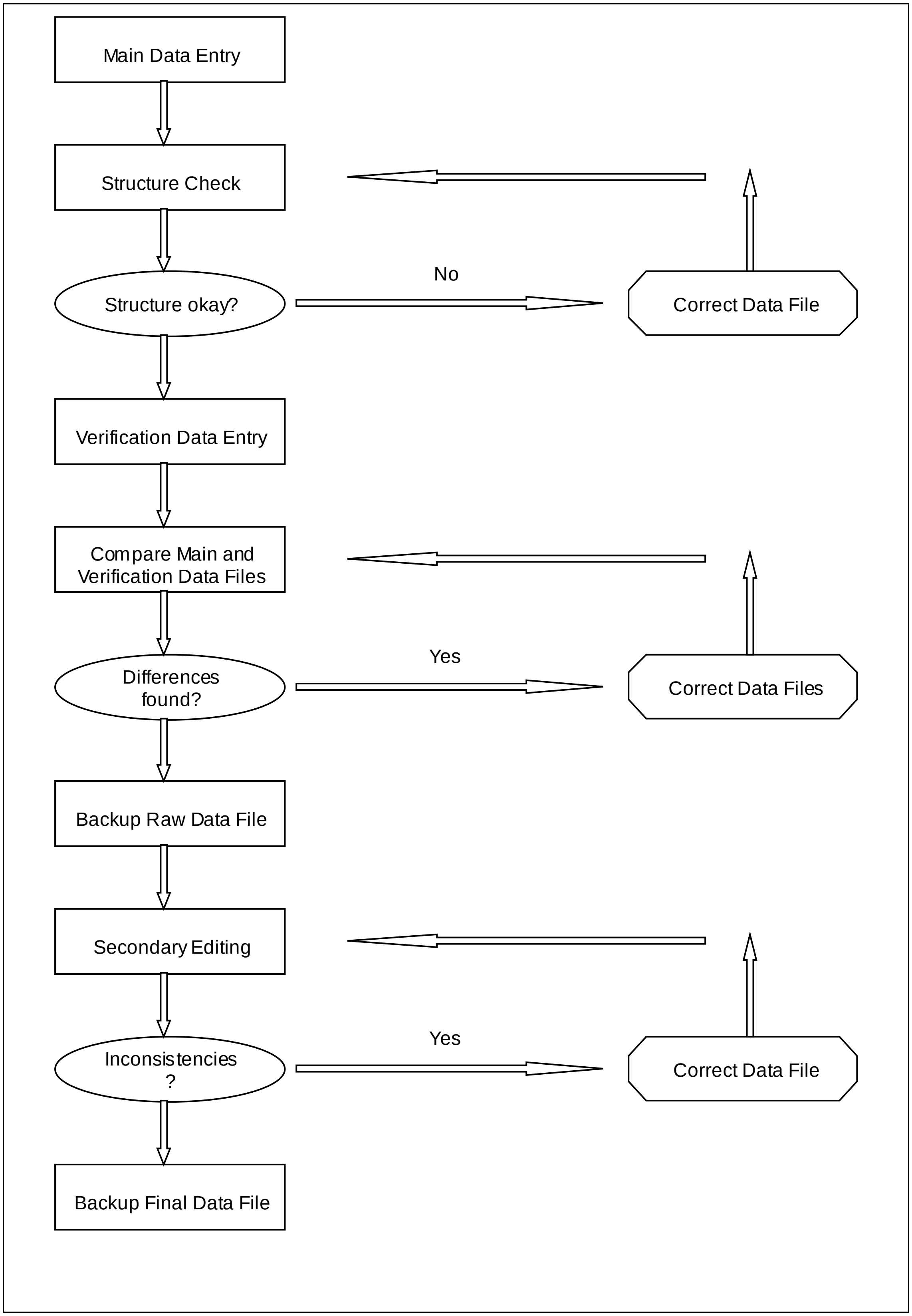
A MICS survey has two data processing phases: primary and secondary. The goal of primary data processing is to produce clean, edited data files. Primary data processing comprises the following steps.

- Data entry
- Structure check
- Verification
- Secondary editing
- Backup of data

The goal of secondary data processing is to produce data files to be used for analysis, including tabulations. Secondary data processing comprises the following steps.

- Calculation of weights
- Recoding of variables
- Tabulation

The flow chart on the following page illustrates the flow of primary data processing.


Main Data Entry
Structure Check
Structure okay?
No
Correct Data File
Verification Data Entry
Compare Main and Verification Data Files
Differences found?
Yes
Correct Data Files
Backup Raw Data File
Secondary Editing
Inconsistencies ?
Yes
Correct Data File
Backup Final Data File

Main data entry

The data for each cluster are entered twice. The first time the data is entered is called main data entry and the second time is called verification data entry. One data entry agent must do the main data entry and another data entry agent must do the verification data entry. Under no circumstances should the same data entry agent enter both the main and verification data.

The data entry program performs basic consistency checks. These checks are designed to catch egregious questionnaire errors and data entry errors. Complex inconsistencies are not addressed until secondary editing.

Structure check

It is essential that all questionnaires be entered and that their identification information be correct. The structure check ensures that both of these requirements are met. Only the structure of the main data entry needs to be checked because any structural errors in the verification data will be discovered during the verification process.

Verification

When main and verification data entry are finished, the two files are compared and a list of differences is produced. The two data entry clerks resolve differences by checking the questionnaires for the correct values. They then correct the values in their data files and the verification program is rerun. The process is continued until the data files are identical. When the verification process is finished, the data should contain exactly what is written on the questionnaires.

Raw data backup

When the data have been verified a backup copy is made before beginning secondary editing. In addition to guarding against accidental data erasure, this backup allows survey personnel to review the changes made during secondary editing. It is recommend that the raw data be backed up to a secondary storage medium (e.g., a zip disk). If this not possible, make a backup copy of the raw data in a separate directory on the hard drive of the supervisor’s machine.

Secondary editing

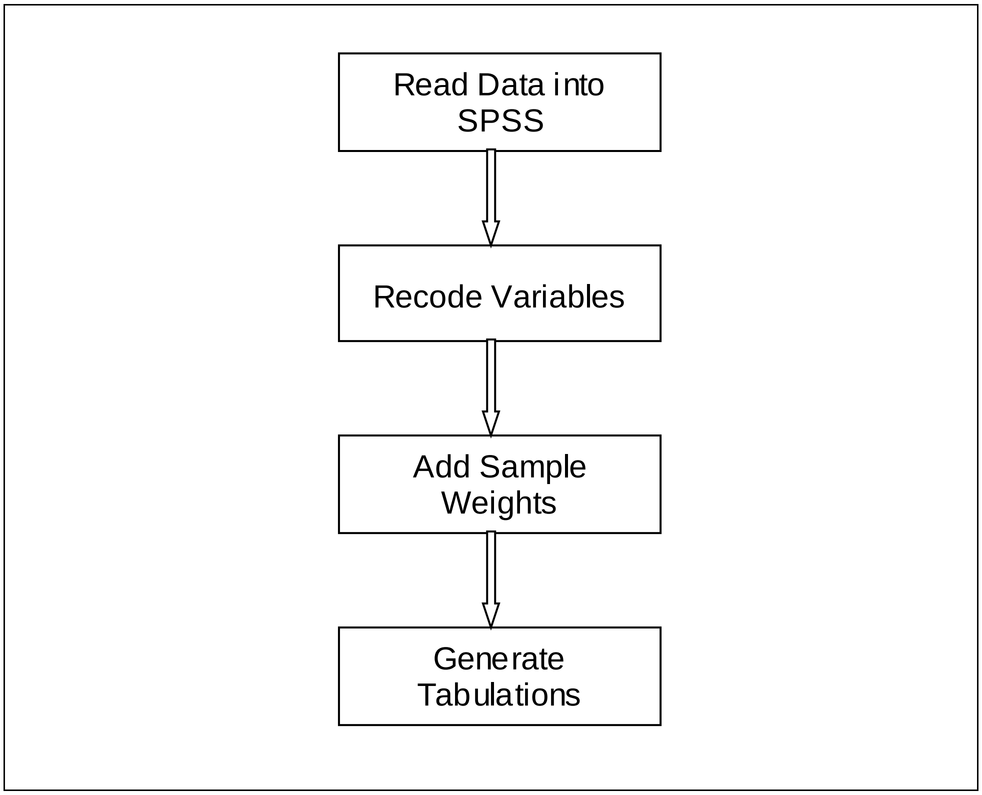
The goal of secondary editing is to investigate complex inconsistencies. The goal is not to systematically correct these inconsistencies. Inconsistencies should only by resolved if they concern essential variables (e.g., ages), if there is clear evidence of interviewer error or if the inconsistency is due to data entry error.

Secondary editing can involve several iterations because resolving one inconsistency often introduces others.

Final data backup

When secondary editing is complete, the final data are backed up. It is recommend that the final data be backed up to a secondary storage medium (e.g., a zip disk). If this is not possible, make a backup copy of the final data in a separate directory on the hard drive of the supervisor’s machine.

The chart below illustrates the flow of secondary data processing.



Concatenating data

The first step in secondary data processing is concatenating all of the cluster files into one national file.

Exporting data

The second step in secondary data processing is exporting the data from the data entry package into SPSS. This includes transferring any variable or value labels.

Recoding variables

Once the data have been exported to SPSS, the process of creating analysis files begins. The first step is to create new variables and to recode existing variables.

Add sample weights

Sample weights are calculated using an Excel spreadsheet. When all the required analysis variables have been created, the sample weights are added to the data files.

Generate tabulations

When analysis files have been created, SPSS programs generate the standard MICS tabulations.

Control Sheet

The control sheet is used to manage the flow of primary data processing. It is stored in an Excel file in the SUPER directory named CONTROL.XLS. The first column contains all of the cluster numbers. Columns B, D, E, G, H, I and J contain space for writing the dates on which the primary data processing tasks are completed. Column C contains space for writing the number of the data entry operator who was assigned main data entry. Column F contains space for writing the number of the data entry operator who was assigned verification data entry.

Before the start of data entry, print a copy of CONTROL.XLS. This will serves as a paper backup of the control sheet. Each time a task is completed, update both the paper copy and the file. The second to last row of the control sheet contains the number of clusters for which each task has been completed. The last row of the control sheet contains the percent of clusters for which each task has been completed. Use these two rows to track the progress of the data processing system.

CONTROL.XLS needs to be modified for your survey. Modify the cluster numbers so that they match the cluster numbers for your survey, adding or deleting rows as necessary. Make sure that cells are correctly formatted (e.g., cells that are to contain dates are defined as date cells). Finally, check the formulas in the last two rows and update them if necessary.

Cluster summary sheet

A cluster summary sheet should be maintained for each cluster of questionnaires. It should be a large, heavy piece of paper with the cluster number written on it in large, bold numerals. The cluster summary sheet should be folded over the end of the package of questionnaires so that the cluster number is displayed clearly. The following fields should be printed on the top of the sheet.

Cluster Number ____________

Total Household Questionnaires __________

Total Households Completed __________

Total Women Questionnaires __________

Total Children Questionnaires __________

The questionnaire administrator should complete the fields when he or she receives the cluster.

Data entry directory structure

The diagram below shows the directory structure on a data entry machine.

```
MICS
      EPIINFO/IMPS/ISSA
            DATA
            ENTRY
            VERIFY
```

EpiInfo/Imps/Issa

This directory, which is named for the data entry package being used, contains the entry menu.

Data

The data directory contains the results of main data entry.

Entry

The entry directory contains the data entry programs.

Verify

The verify directory contains the results of verification data entry.

Supervisor directory structure

The diagram below shows the directory structure on the supervisor's machine.

```
MICS
      EPIINFO/IMPS/ISSA
            BACKUP
            EXPORT
            FINAL
            RAW
            SPSS
            SUPER
            WEIGHTS
```

EpiInfo/Imps/Issa

This directory, which is named for the data entry package being used, contains the supervisor menu.

Backup

The backup directory contains a backup of data files that have been structurally checked and verified but not edited.

Export

The export directory contains the programs used to export the data from the data entry package to SPSS.

Final

The final directory contains a backup of data files that have been structurally checked, verified and edited.

Raw

The raw directory contains the data files that have been transferred from the data entry machines.

Spss

The spss directory contains the programs that create analysis files and the tabulation programs.

Super

The super directory contains the programs that do structural checks, verification and secondary editing.

Weights

The weights directory contains the spreadsheet that calculates sample weights.

Data description

There are two files involved in creating a data description for EpiInfo. The first file is a questionnaire file and is referred to as a QES file (its filename ends with a QES extension). This file is a layout of the questionnaire that is to be entered.

When data entry begins for the first time, EpiInfo converts this file into a REC file (its filename ends with an REC extension). The record file begins with a description of the data and the forms to be used during data entry. The remainder of the file is the data that are entered. Once the QES file has been converted, it is no longer used.

There are three types of MICS questionnaires: household, women and children under five. The women's and children's questionnaires each 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's member.

Each of the questionnaire types is stored in a separate file and each file uses a separate data description (often called a dictionary). Because EpiInfo does not permit repeating variables, it stores information about household members – the household listing, education and child labour modules – in a separate file. This file also has a data description. The data descriptions are named as follows.

Questionnaire	Data Description
Household	MICSHH
Household listing	MICSHL
Women	MICSWM
Children	MICSCH

EpiInfo therefore has four QES files and four REC files.

Modifying the data description

REC files can be modified directly, but their syntax is complicated and it is easy to make a mistake. The standard MICS REC files were created by converting QES files. It is recommended that you modify the data description by changing the QES file and then converting it into a REC file.

Modules

The MICS questionnaires are made up of modules. Countries can add or remove modules, depending on their data needs. Countries are discouraged from significantly altering the contents of a module. The modules available for each questionnaire are listed below.

Questionnaire type	Module	Code
Household	Household information	HI
	Household listing	HL
	Education	ED
	Child labor	CL
	Water and sanitation	WS
	Salt iodization	SI
Women	Women's Information panel	WI
	Child mortality	CM
	Tetanus toxoid	TT
	Maternal and newborn health	MN
	Contraceptive use	CU
	HIV/AIDS	HA
Children	Birth registration and early learning	BR
	Vitamin A	VA
	Breastfeeding	BF
	Care of illness	CI
	Malaria	ML
	Immunization	IM
	Anthropometry	AN

EpiInfo does not allow more than one record per case, so modules in EpiInfo are represented by spaces between the forms in the QES file. The QES file is aligned so that the beginning of each module will be displayed at the top of the screen.

Each screen in EpiInfo displays 23 lines. You can calculate the line number of the first line of each screen by adding 23 to the first line on the preceding screen (i.e., 1, 24, 47, 70, 93, 116, 139, etc.). Whenever you begin a new module in the QES file, make sure you are on a line that will be displayed at the top of the screen.

Variable naming conventions

Variables are named for the module that they are in and the number of the question whose response they contain. For example, question 8 in the household listing is named HL8.

Some questions are split into two or more parts, with each part being prefaced by a letter. Each part of the question is a separate variable. The names of these separate variables include the letters that distinguish the parts of the question. For example, question 3 of the women's information panel has two parts, 3A and 3B. Question 3B is named WI3B.

Some questions have two or more parts to the response categories. When these questions concern data, the letters D (for day), M (for month) and Y (for year) are appended to the base variable to create separate variables. In question 3A of the women's information panel, for example, both a month and a year are required. These two variables are named WI3AM and WI3AY.

There are two questions in the questionnaires in which the first part of the response is the form of the response and the second part is the response: question 6 in the maternal and newborn health module of the women's questionnaire and question 3 of the anthropometry module of the children's questionnaire. In both cases, the variable giving the form of the response has the letter A appended to its name while the variable containing the response has no letter appended (e.g., MN6A and MN6, AN2A and AN2).

There are a number of questions in the questionnaire that allow for multiple responses. For these questions, each response has its own variable. The variables' names are made up of the module code, the question number and the letters A through K, which correspond to the response codes 1 to 11. An exception is the response "don't know" which uses the letter Z in all cases. For example, question 4 of the malaria module in the children's questionnaire has variables named ML4A, ML4B, ML4C, ML4D, ML4Z.

Identification variables

Every questionnaire must have a series of variables that uniquely identifies it. The variables that identify a questionnaire are known as the identification variables. The table below lists the questionnaire types and their identification variables.

Questionnaire type	Cluster number	Household number	Line number
Household	HI1	HI2	
Household listing	HI1	HI2	HL1
Women	WICLNO	WIHHNO	WILNNO
Children	CHCLNO	CHHHNO	CHLNNO

In addition, each of the data descriptions contains compound identification variables. These variables are necessary because EpiInfo uses one variable only when executing the RELATE command. The table below shows the compound variables each data description contains and the questionnaires to which they provide a link.

Questionnaire type	Household	Household listing	Women	Children
Household		HID	HID	HID
Household listing	HID		HHMID	HHMID
Women	HID	HHMID		
Children	HID	HHMID		

The variable HID is comprised of cluster number and household number. It uniquely identifies every household. HHMID is comprised of cluster number, household number and line number. It uniquely identifies every household member. It therefore also uniquely identifies every woman and every child.

MICSHH.QES

The household QES is unremarkable except for five added variables: HID, HIMEM, WOMEN, CHILDREN and FINISHED. These variables do not exist on the MICS standard questionnaires. They have been added to increase control over data entry.

HIMEM is used in the household listing data entry program to ensure that the correct number of household members is added. WOMEN, CHILDREN and FINISHED are used to control the entry of women's and children's questionnaires. All four variables will be examined in detail when the data entry programs are examined.

MICSHL.QES

The household listing QES contains four added variables: HI1, HI2, HID and HHMID. These variables have already been discussed in the identification variables section.

The household listing QES has another notable feature: it variables are arranged in columns rather than in rows. This arrangement reflects the tables on the questionnaire. Notice that each line of variables ends with the characters

```
{o}
```

This character, which defines a field named O, is necessary because of EpiInfo's automatic filed name feature. EpiInfo assumes that any line with characters is a field and assigns it a name. If no alphabetic characters exist on the line, EpiInfo will name it by incrementing the name of the previous field.

For example, consider the 12th line of MICSHL.QES. The last field on this line before the O field is HL7. The two lines that follow it, lines 13th and 14th, contain characters but no alphanumeric characters. If the

```
{o}
```

character were not present, EpiInfo would name the fields created from these lines HL8 and HL9. These names would conflict with the legitimate HL8 and HL9 fields, so EpiInfo would rename the legitimate fields HL801 and HL901. This would cause obvious problems for the data entry application!

However, the

```
{o}
```

character is present, so EpiInfo names these two lines O1 and O2. You must be aware of this problem anytime you work with a QES file that contains lines with characters but no fields.

MICSWM.QES

The women's QES file contains two added variables that we have already discussed: HID and HHMID. It also contains an added variable that we have not discussed: CM11X. CM11X reproduces the filter at the bottom of the child mortality module. Without it, it is impossible for the data entry program to correctly follow the questionnaires skip pattern.

MICSCH.QES

The children's QES file, like the women's QES file, contains the added variables HID and HHMID. In addition, the anthropometry section of the children's questionnaire contains 4 additional variables – SEX, DOID, DOIM and DOIY. These variables are required for the calculation of anthropometry. The data for the 4 fields are not listed on the children's questionnaires; the data entry operators must obtain this information from the household questionnaire.

Data entry programs

EpiInfo's data entry programs are known as check files (their filenames have a CHK extension). They are used to check ranges, check basic consistency, control the data entry path (the path that a data entry operator follows through the questionnaire) and link related files.

Coding conventions

The standard programs use standard coding for certain responses. The response "don't know" is always coded as a nine with leading nines. Questions that are not applicable to a respondent are always coded as an eight with leading nines. Questions with a missing response (i.e., the interviewer did not record a response to an applicable question) are always coded as a seven with leading nines. Responses that are inconsistent with other data in the questionnaire are always coded as a six with leading nines. The table below summarizes the standard coding conventions.

Response	Variable length			
	One character	Two characters	Three characters	Four characters
Don't know	9	99	999	9999
Missing	7	97	997	9997

All the response categories for a variable should contain the same number of digits. If a question requires two digits, codes with values less than 10 must have leading zeros. For example, if a question uses codes 1 through 12, the codes on the questionnaire must be 01-12 and not 1-12.

Because the codes 7, 8 and 9 are reserved for special use, any question that requires more than 6 response categories must have 2-digit responses categories with leading zeros (e.g., 01, 02, 03, 04, 05, 06, 07, 97 and 99).

Ranges

Most of the variables in the MICS questionnaires have defined ranges. Only values within the specified ranges can be entered. This reduces the probability of data entry errors. The only variables without defined ranges are variables for which it is impossible to create restrictive ranges. The limits of household number, for example, vary so much from cluster to cluster that any defined ranges would catch only a small number of errors.

EpiInfo allows an infinite number of ranges.

Skips

The MICS 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 help define the data entry path.

EpiInfo does not strictly enforce the data entry path. Data entry operators can enter and change fields that are not on the data entry path. They can also skip fields in which they should enter data. This problem can be controlled using the MUSTENTER keyword, but MUSTENTER introduces a problem of its own: it complicates moving backward through the questionnaire. MUSTENTER is only used for critical fields (e.g., identification fields) in the standard programs.

Throughout the data entry programs you will see the following code.

```
IF VAR = . THEN
  GOTO VAR
ENDIF
```

This code forces the data entry operator to enter a value in a field. If no value is entered, EpiInfo will skip back to the question and the data entry operator will be unable to advance.

Error messages

Error messages are displayed using the HELP command. The error messages are numbered and a master list of the error message number exists. If you add error messages, make sure that you do not reuse a number.

Many of the error messages are followed by a skip that returns to the field that is being entered. This forces the entry operator to address the error before advancing.

Multiple response questions

There are a number of questions in the questionnaire that allow for multiple responses. For these questions, each response has its own variable. Each variable has three possible responses – 0 for no, the responses code on the questionnaire and 7 for missing data.

For example, the variable MN2 records whom a woman saw for antenatal care before her last birth. The response code 2 is associated with the response nurse/midwife. The variable for this response is named MN2B. The allowed entries are 0, 2 or 7.

If a missing value is entered for any of the responses, a missing value must be entered for all of the responses. The number of missing values must therefore equal zero or the number of responses.

This requirement is checked using the working variable MISS. MISS is initialized to zero and then incremented by one if a missing value is entered for a response. When the last response has been entered, MISS must equal either zero or the number of responses. If it does not, an error message is displayed and the entry operator is unable to advance until the inconsistency has been resolved.

Color schemes

Each of the questionnaires is displayed with its own background color (the text is always yellow).

Program	Background color
MICSHH	Blue
MICSHL	Black
MICSWM	Red
MICSCH	Green

The different colors help data entry operators distinguish between the different questionnaires.

MICSHH.CHK

The household data entry program launches all of the other data entry programs. In spite of this, it is neither long nor complex. Understanding how it works is essential to understanding the MICS data entry programs.

Throughout this section we will refer to blocks of code related to a specific variable as procedures. Thus, the code related to the variable HI1 is the cluster number procedure. Only procedures that contain unusual or complex code will be discussed.

BEFORE FILE procedure

The working variables CLUSTER and HHNUM are created to store the questionnaire's cluster and household number. They will be used to control these values on the related household listing, women and children records. HHNUM is initialized to the value 0 to prevent the consistency check in the household number procedure from generating an error message for the first household in a cluster.

BEFORE RECORD procedure

Two important variables are created in this procedure. The first, HHMAX, is used in MICSHL.CHK to control the number of household listing records. The second, MISS, has already been examined.

HI1 procedure

When the data entry program is launched, a one-line file is created which contains the current cluster number. This file, CL.TXT, is then used to define the legal values for the HI1 field. This ensures that the correct cluster number is entered.

HIMEM procedure

The HIMEM procedure launches data entry for the household listing using the RELATE command. Before doing so, it initializes the values of HHMAX, CLUSTER and HHNUM.

WOMEN procedure

This procedure controls the entry of women's questionnaire. If the data entry operator enters the value Y, a RELATE command launches data entry of the women's questionnaire. A value of N moves on to the next field without entering any women's questionnaires.

CHILDREN procedure

This procedure controls the entry of children's questionnaire. If the data entry operator enters the value Y, a RELATE command launches data entry of the children's questionnaire. A value of N moves on to the next field without entering any children's questionnaires.

FINISHED procedure

This procedure requires the data entry operator to explicitly finish data entry for the current household. Data entry will only be ended if the data entry operator enters the value Y.

MICSHL.CHK

During data entry, the household data entry program launches the household listing data entry program. During data modification, the household listing data entry program is executed as a stand-alone program to speed up the modification process. Much of the remarkable code in MICSHL.CHK results from this functionality.

BEFORE FILE procedure

The first two lines of the procedure set the household listing's color scheme. The last two lines define the working variables CLUSTER and HHNUM. If MICSHH.CHK has launched MICSHL.CHK, these definitions are redundant and EpiInfo ignores them.

BEFORE RECORD procedure

The first part of this procedure defines the working variables HHLN and HHMAX. If MICSHH.CHK has launched MICSHL.CHK, these definitions are redundant and EpiInfo ignores them.

The second part sets the values of HI1 and HI2 if MICSHH.CHK launched MICSHL.CHK. Otherwise it sets the value of HHMAX to 50, the maximum number of household members.

The third part increments the value of HL1 (line number) by 1 and sets the value of HHMID (household member ID).

HI1 procedure

The portion of this procedure that is executed before entry skips to the field HL1 if MICSHH.CHK has executed MICSHL.CHK. For these case, the HID field has already been completed by the RELATE command.

HL1 procedure

The code

```
IF HL1 > HHMAX THEN
  HELP "0081 DATA ENTRY IS FINISHED FOR THIS HOUSEHOLD." 1 1
GOTO HL1
ELSE
  HHMID = HI1*100000 + HI2*100 + HL1
  ED14  = HL1
  CL1   = HL1
  HHLN  = HL1
ENDIF
```

forces the data entry operator to stop data entry if the current line number is greater than the number of household members.

If the current line number is valid, the procedure sets the value of the line number variables ED14 and CL1 equal to the current line number. The procedures of fields ED14 and CL1 both contain the NOENTER command. The procedure also updates the value of HHLN, the variable that stores the current line number.

The second line has been truncated because of lack of space.

MICSWM.CHK

During data entry, the household data entry program launches the women's data entry program. During data modification, the women's data entry program is executed as a stand-alone program to speed up the modification process. All of the remarkable code in MICSWM.CHK results from this functionality.

BEFORE FILE procedure

The first two lines of the procedure set the women's questionnaire's color scheme. The last two lines define the working variables CLUSTER and HHNUM. If MICSHH.CHK has launched MICSWM.CHK, these definitions are redundant and EpiInfo ignores them.

BEFORE RECORD procedure

The first part of this procedure sets the values of WICLNO and WIHHNO if MICSHH.CHK launched MICSWM.CHK. The second part defines working variables.

MICSCH.CHK

During data entry, the household data entry program launches the children's data entry program. During data modification, the children's data entry program is executed as a stand-alone program to speed up the modification process. Much of the remarkable code in MICSCH.CHK results from this functionality.

BEFORE FILE procedure

The first two lines of the procedure set the children's questionnaire's color scheme. The last two lines define the working variables CLUSTER and HHNUM. If MICSHH.CHK has launched MICSCH.CHK, these definitions are redundant and EpiInfo ignores them.

BEFORE RECORD procedure

The first part of this procedure sets the values of CHCLNO and CHHHNO if MICSHH.CHK launched MICSCH.CHK. The second part defines working variables.

DOIY procedure

The complex code in this procedure 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. Calculating a child's age in days requires a reference point that is before the data of birth of all possible births. In this case the 1st of January of 1995 was chosen as the reference point.

The code first calculates the number of days that elapsed between the reference point and a child's birth. It then calculates the number of days that elapsed between the reference point and the date of interview. The difference between these two numbers of days is the child's age in days. This is then convert 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' age in months is calculated to two decimal places.

Structure checking

Structure checking is an essential part of the MICS data processing approach. It is essential that the data be structural sound. The data entry programs do not enforce structural coherence (there are several reason for this, but the most important is that it would constrain data entry too much) so a structural checking program must do so.

Assumptions

Before examining the structure checking program, consider the lists below that layout some of the structural assumptions that are being checked. The lists are not exhaustive, but they do illustrate the general principles on which structural checks are based.

Household

- The number of households should equal the sum of the number of incomplete and complete households.
- The number of records in the household data file should equal the number of households implied by the household listing data file.

Household listing

- The number of household members listed on the household cover sheet should equal the number of household member records in the household listing data file.
- The number of household members who are women age 15 to 49 should match the number of eligible women on the household cover sheet.
- The number of household members who are children under the age of 5 should match the number of eligible children on the household cover sheet.

Women

- The number of women's records should match the number of interviewed women on the household cover sheet.
- There should be a one-to-one correspondence between records in the women's data files and records in the household listing data file for women age 15 to 49.

Children

- The number of children's records should match the number of completed children interviews on the household cover sheet.
- There should be a one-to-one correspondence between records in the children's data files and records in the household listing data file for children under the age of 5.

Notice that many of the assumptions above are true for both the household and the cluster. For example, the number of eligible women on the household cover sheet should equal the number of female household members age 15 to 49. At the same time, the sum of eligible women for all households in the cluster should equal the number of female household members age 15 to 49 for the entire cluster.

Programs

Structure checking is done using EpiInfo’s analysis module. The structure checking program, CHECK.PGM, is split into 3 parts (comments in the program mark the beginning of each part). Each part produces its own output and serves a distinct purpose. Together, the three parts check the structural coherence of the household, household listing, women’s and children’s data files.

The workings of the three parts will not be explained here in detail. We will focus on what the programs are doing instead of how they do it. The programs are complex and do not lend themselves to facile explanations. The best way to understand them is to study them carefully when you understand what they are doing.

Part one

The first part of CHECK.PGM counts the number of questionnaires expected and compares the results to the number of questionnaires in the data files. The output below shows an example based on one household.

```
                          MICS Data Structure Check
Cluster:   3

    Households     |Household Members|     Women        |     Children
Total  Comp  Incomp|  Cover    Found | Elig. Int. Found | Elig. Int. Found
   1     1      0  |     6       5   |   2     2     2  |   0     0     1

** Total household members incorrect
** Total children interviewed incorrect
```

There are two structural problems in this example. First, the household cover sheet records 6 household members but only 5 household listing records exist in the data file. Second, there is a case in the child’s data file, but the household cover sheet records no children under the age of 5 and no children’s interviews.

These errors can be seen from the summary table and from the two error messages.

Part two

The second part of CHECK.PGM provides a summary of households and their members. Information about the household is displayed first followed by information about the household’s members, if any. Information is displayed about all households in the data files, even if they are structurally correct. The output below continues the one household example used in part one.

```
Household     HI10 |  HIMEM          |HI11    HI12      |HI13    HI14
Household member   |            HL1  |  HL5       WILNNO|  HL7       CHLNNO

     3001   res=1  |     6           | 2       2        | 0       0
     300101        |             1   |   0           .  |   0           .
     300102        |             2   |   2           2  |   0           .
     300103        |             3   |   3           .  |   0           .
     300104        |             4   |   0           .  |   0           .
     300105        |             5   |   0           .  |   0           5
** 0210 Interviewed child not eligible
** Number of household members incorrect
** 0209 Number of interviewed women incorrect
** 0208 Number of interviewed children incorrect
```

HI10 is the result of the household interview. In the example, the household interview was completed. HIMEM is the number of household members. Notice that it is followed by only 5 household members; the household member will line number 6 is missing.

HI11 is the number of eligible women. HL5 is a household variable that reflects eligibility for the women's interview. Two household members are recorded as eligible, a total that matches HL5. HI12 is the number of interviewed women. WILNNO is the line number for a record in the women's data file. Only one household member has a value meaning that one women's questionnaire is missing for this household, one women's questionnaire for this cluster has an incorrect line number or the value of HI12 is incorrect.

HI13 is the number of eligible children; HL7 is the line number of an eligible child's caretaker. HI14 is the number of complete children's interviews. CHLNNO is the line number for a record in the children's data file. One such record exists despite the value of HI14 (0) meaning that the eligibility code of one of the household members is incorrect and the values HI13 and HI14 are incorrect or the household identification for the child's questionnaire is incorrect.

The four messages that follow the household summary summarize the errors mentioned above. Messages one and four derive from the same source.

Part three

Part three is similar to part one. The difference is that part one's counts are based on what is present in the data files while part three's counts are constrained by some basic consistency checks. The output below continues the one household example used in part one.

```
    Households     |Household Members|     Women        |     Children
Total  Comp  Incomp|  Cover   Listed | Elig. Int. Found | Elig. Int. Found
   1     1      0  |     6       5   |   2     2     1  |   0     0     1

** Total household members incorrect
** Total women interviewed incorrect
** Total children interviewed incorrect

** Total women's questionnaires inconsistent
```

The difference between these two approaches can be seen in the found column of the women's section of the table. Part one found 2 women; part three found 1 woman. Part one counted the number of women's records in the women's data files. Part three counted the number of household listing records who had a matching record in the women's data file.

The missing women's questionnaire had an incorrect line number: 9 instead of 3. Because no household listing record has a line number of 9, this women's record could not be matched to a household listing record and was not counted. The error messages that follow the table summarize the errors that can be observed in the table.

Note that the totals that are used in part three are computed in part two; only the output is created in part three.

Timing

A critical feature of the structure checking program is the timing of the execution of the statements. The structure checking program works on a case-by-case basis. All statements are executed for one case. This contracts with statement-by-statement basis in which one statement is executed for all cases. The case-by-case timing of execution is essential to the proper functioning of the structure checking program.

The structure checking program achieves this timing through the use of IF statements. These statements are executed on a case-by-case basis. However, they are not executed when they are encountered; they are executed when EpiInfo encounters a PROCESS statement. The

PROCESS statement tells EpiInfo to execute all of its built up statements. By combining IF and PROCESS statements, we have forced EpiInfo to treat cases on a case-by-case basis.

In the program, you will encounter a statement in which the condition is whether zero is equal to zero. This is a command that must be executed for all cases; the IF statement (and its self evident condition) is there to maintain the case-by-case timing. Because the condition is true, the condition is always executed; because an IF statement is used, the timing of execution is preserved.

Read through the program and pay careful attention to the timing. Keeping the output of the structure checking program in mind should help you to understand why the program is moving case-by-case. If you make modifications to the program, precede all statements with an IF statement unless you are certain that you wish to change the timing. If there is a statement that should be executed for all cases, use the zero equals to zero condition mentioned above.

Verification

Verification is done using EpiInfo's validate module. The main data entry file (which has been copied onto the supervisor's machine) is compared to verification data entry file (which has been copied onto a diskette). Validate will produce a list of differences if any exist. If there are differences, this list should be printed out and given to the two data entry operators.

The data entry operators then sit down with the list and the cluster's questionnaires. They must check each difference against the questionnaire and record 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 and the files compared again. This process continues until the files are identical.

Secondary editing

The goal of secondary editing is to catch inconsistencies in the data that have not, and should not, be resolved during data entry. Some inconsistencies must be corrected while others are only corrected if a data entry error occurred. The secondary editors must carefully investigate the error listing produced by the editing process.

Their investigation of the error listing and all corrections are dictated by the editing guidelines. The editing guidelines contain a list of every message, information on what actions to take and hints about possible causes of the inconsistency. Because the editing guidelines are thorough and the editing programs are straightforward, we will only briefly discuss the editing process.

Secondary editing is done using EpiInfo's analysis module. There are four secondary editing analysis programs: EDITHH.PGM, EDITWM.PGM, EDITCH1.PGM and EDITCH2.PGM. To produce an error listing organized by case, the editing programs processes cases one at a time. To do this, it relies on the same timing of execution as the structure checking programs. The editing programs therefore consist mainly of IF and PROCESS statements.

This approach uses a lot of memory, so the editing programs have been split into four parts: one each for the household and women's data files and two for the children's data files. However, the four programs are executed together by the supervisor's batch file and all write to the same error listing.

Menu systems

Two menus manage the MICS data processing system. Data entry operators use the data entry menu to enter, modify and transfer data. The data processing supervisor uses the supervisor menu to transfer and verify data, execute the structure checking and editing programs and to backup data files.

Both menus are implemented using EpiInfo's menu system.

Data entry menu

The diagram below shows the contents and structure of the data entry menu. Typing ENTRY in the C:\MICS\EPIINFO directory launches the menu system.

```
Entry
      Cluster
            Cluster number
            QUIT
      Enter
            Enter Main Data
            Enter Verify Data
      Modify
            Modify Main Data
                  Household Data
                  Household Listing Data
                  Women’s Data
                  Children’s Data
            Modify Verify Data
                  Household Data
                  Household Listing Data
                  Women’s Data
                  Children’s Data
      Transfer
            Transfer Main Data
            Transfer Verify Data
      Setup
            Main Data Directory
            Verification Data Directory
            Program Directory
            Transfer Diskette Directory
```

Cluster

The cluster menu contains two options: cluster number and quit. The cluster number allows the data entry operator to set the current cluster. The data entry operator should always select a cluster number before doing anything else.

The cluster number is selected from the ASCII text file CLUSTER. Replace the list of cluster numbers in the standard file with the list of cluster numbers in your country. Use leading zeros so that each cluster number has the same number of characters.

The quit option exits the data entry menu.

Enter

The enter menu allows the data entry operator to enter main or verification data. The standard data descriptions are copied from the ENTRY directory to data directory (DATA or VERIFY) are renamed for the current cluster.

Modify

The modify menu allows the data entry operator to modify the main or verification data for any of the four data file types. Note that modifying the household data will also allow the data entry operator to modify the other data types. The household listing, women's and children's data files can be modified individually.

Transfer

The transfer menu copies data files from the data directories onto a diskette. The data files are then copied onto the supervisor machine using the supervisor menu. The diskettes onto which the data files are copied also serve as backups of the data on the data entry machines.

Setup

The setup menu allows the user to establish the main data, verification data, program and transfer directories. Once your data processing system has been successfully installed, you may wish to remove this menu to prevent data entry operators from changing directories.

Supervisor menu

The diagram below shows the contents and structure of the supervisor menu. Typing SUPER in the C:\MICS\EPIINFO directory launches the menu system.

```
Super
      Cluster
            Cluster number
            Transfer Cluster Data Files
            QUIT
      Check
            Main Data
      Verify
            Household Data
            Household Listing Data
            Women’s Data
            Children’s Data
      Edit
            Main Data
      Modify
            Household Data
            Household Listing Data
            Women’s Data
            Children’s Data
      Backup
            Backup Verified Data Before Editing
            Backup Final Data After Editing
      Setup
            Raw Data Directory
```

```
Backup Data Directory
Final Data Directory
Program Directory
Output Directory
```

Cluster

The transfer cluster data files option allows the data processing supervisor to transfer the main data for a cluster from a diskette into the RAW data directory. The verification data stays on a diskette. The cluster number and quit options are the same as on the data entry menu.

Check

The check menu allows the data processing supervisor to check the structure of the main data files. Any output is displayed using EpiInfo’s VIEW feature.

Verify

The verify menu allows the data processing supervisor to verify the main and verification data files. The list of differences, if any, is displayed using EpiInfo’s VIEW feature.

Edit

The edit menu allows the data processing supervisor to run the editing program on the main data files. The list of errors, if any, is displayed using EpiInfo’s VIEW feature.

Modify

The modify menu allows the supervisor to modify data files. This option is used to correct errors encountered during the editing process.

Backup

The backup menu allows the data entry supervisor to backup data files.

Setup

The setup menu allows the data entry supervisor to set the raw data, backup data, final data, program and output directories. Once the data processing system has been successfully installed, you may wish to remove this menu.

Exporting data

Concatenation

When data entry is complete, four final (i.e., verified and edited) data files will exist for each cluster: one for households, one for household members, one for women and one for children. These cluster data files must be concatenate into four national files.

The files can be concatenated using the MERGEHH.BAT, MERGEHL.BAT, MERGEWM.BAT and MERGECH.BAT (collectively known as the MERGE batch files) batch files. The batch files concatenate the cluster data files using the EPIINFO merge program. Consider the example below from MERGEHH.BAT.

```
merge hht006 hh010m hht010 1
```

The first two parameters following the merge command are the files to be merged. The third parameter is the new, merged file. The fourth parameter, the number 1, specifies that the files should be concatenated.

The merge program calls in the MERGE batch files are arranged so that the numbers in the file names can be replaced with the list of cluster numbers in the CLUSTERS file that was used during data entry.

To use one of the merge batch files

1. Open an MS-DOS window and navigate to the EXPORT sub-directory (e.g., \MICS\EPIINFO\EXPORT)
2. Type the name of the batch file and press the enter key

The merge commands are followed by the MS-DOS command rename, which renames the final merged data file. The example below is from MERGEHH.BAT

```
rename hht448.rec hh.rec
```

Exporting the data

The MERGE batch files also export the concatenated data files that they produce into SPSS. They do this using the EPIINFO export program. Consider the command below from MERGEWM.BAT

```
export wm readwm 11
```

The first parameter following the export command is the REC file to be exported. The second parameter is the output file. The third parameter is the package of the output file (11 is SPSS/PC). Notice that file extensions are not allowed on the first and second parameters. EPIINFO assumes that the file to be exported is a REC file. The file extension for the output file is determined by the third parameter.

Output

When the MERGE batch files are executed, they will produce four SPSS data description files: READHH.SPS, READHL.SPS, READWM.SPS and READCH.SPS (collectively known as the READ files). The data description files contain both the data and their description.

The SPSS command

```
      save outfile = ‘filename.sav’.
```

must be modified and added to the end of each data description file. The word FILENAME must be replace with HH, HL, WM or CH, depending on the type of data file. This command will save the data file that is created by each syntax file.

Once this command has been added to each data description file, executing the READ files in SPSS will create the SPSS data files HH.SAV, HL.SAV, WM.SAV and CH.SAV.

Creating Analysis Files

Adding labels

Before using a data file, it is helpful to add variable and value labels. A variable label describes the subject matter of a variable. Value labels describe what each value of a variable represents.

Consider the household listing variable HL3 that stores the sex of household members. It takes a value of 1 for males, 2 for females, and 7 for missing data. The variable should be labeled “Sex”. The value 1 should be labeled “Male”. The value 2 should be labeled “Female”. The value 7 should be labeled “Missing”.

The SPSS programs LABELHH.SPS, LABELHL.SPS, LABELWM.SPS and LABELCH.SPS (collectively known as the LABEL programs) assign variable and value labels to all the variables in the standard questionnaire. If you have added variables to the questionnaire, go through these programs and add variable and value labels where appropriate.

The LABEL programs are unnecessary if you are using ISSA; ISSA transfers variable and value labels when it exports data.

Adding variables from other files

The SPSS program MAKEHL.SPS, MAKEWM.SPS and MAKECH.SPS (collectively known as the MAKE programs) perform two functions: they add variables from other files and they recode variables to create new variables. Each of these functions will be dealt with in separate sections. This section will concern itself with adding variables from other files.

When analyzing household members, it is useful to have access to variables on the household questionnaire (e.g., HI6, HI7). These variables can be added to the household listing data file by merging it the household data file. More broadly, useful variables can be added to the household listing, women’s and children’s data files from other MICS data files. The table below shows these three data files and the files with which they are merged.

Base file	External file
HL.SAV	HH.SAV
WM.SAV	HL.SAV
CH.SAV	WM.SAV

These merges are executed in the order implied by tables. As a result, when the household listing file is added to the women's file, it adds both household and household listing variables. Similarly, when the women's file is added to the children's file, it adds household, household listing and women's variables.

SPSS Macros

Macros in SPSS are used to automate a series of statements that are used repeatedly. The MAKE programs use macros to simplify the process of merging files. MAKEWM.SPS and MAKECH use the same macro.

The code below is the macro used in MAKEWM.SPS and MAKECH.SPS.

```
*define a MACRO that merges 2 files with 3 ID variables.
define !merge (file1 = !tokens(1)
               /mvar1 = !tokens(1)
               /mvar2 = !tokens(1)
               /mvar3 = !tokens(1)
               /file2 = !tokens(1)
               /name1 = !tokens(1)
               /name2 = !tokens(1)
               /name3 = !tokens(1)).
get file = !file1.
sort cases by !mvar1 !mvar2 !mvar3.
save outfile = 'tmp.sav'
  /rename  =  (!mvar1  =  !name1)  (!mvar2  =  !name2)  (!mvar3  =
!name3).
get file = !file2.
sort cases by !name1 !name2 !name3.
match files
  /file = *
  /table = 'tmp.sav'
  /by !name1 !name2 !name3.
save outfile = !file2.
get file = !file2.
erase file = 'tmp.sav'.
!enddefine.
```

The define keyword begins the macro and the enddefine keyword ends it. The macro's name, merge, immediately follows the define keyword and is preceded by the exclamation point character.

The text enclosed in parenthesis that follows the macro's name defines accepted parameters. For example, file1 and file2 are parameters that represent file names. When a parameter is referred to in the main body of the macro, it is preceded by the exclamation point. The remaining statements in the macro are standard SPSS commands that will not be discussed here.

The statements above define the macro but do not execute any of the commands within. The commands within are only executed when the macro is called.

A macro is called by typing its name preceded by the exclamation point. When it is called, any parameters used within the macro must be specified. The following lines show the calling of the MERGE macro in the MAKEWM.SPS program.

```
*add variables from the household listing file.
```

```
!merge  file1=hl.sav  mvar1=HI1  mvar2=HI2  mvar3=HL1  file2=wm.sav
name1=WICLNO name2=WIHHNO name3=WILNNO.
```

This statement will merge the files HL.SAV and WM.SAV. The files will be merged using the cluster number, the household number and the woman's line number.

Warning!

The merge above is of a keyed table (e.g., HL.SAV) onto a file (WM.SAV). With this type of merge, extra cases in the keyed table (e.g., household members who are not women aged 15-49) are not added to the file. In other types of merges, these extra cases would be appended to the end of the file. If you modify the match files statement, do not replace the /table specification with a /file specification. This will result in ineligible household members being added to the women's data file. This warning applies to all of the MAKE programs, not just the MAKEWM.SPS program.

Recoding variables

Variables used in the standard tabulation plan are recoded in two places: the MAKE programs and the tabulation programs. Variables that are used in several tabulations are recoded in the MAKE programs. All other variables are created in the tabulation programs.

The recoding of most variables uses standard SPSS commands and will not be discussed here. There are however a few unusual approaches that will be explained.

One of these is the recoding of variables into 0 or 100. This unusual recoding is down for presentation purposes only. When SPSS displays percentages in a table, it displays all 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.

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. In the first column of table 30, we want to display only the percentage of women who have heard of AIDS.

In the program MAKEWM.SPS, the variable HA1 is recoded into the variable AIDS. AIDS takes a value of 100 if the woman has heard of AIDS and a value of 0 otherwise. The mean of AIDS is the percent 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 * 100 or 50%**
Mean of the variable AIDS	(10*100 + 10*0) / 20
	10 * 100 / 20
	10 / 20 * 100 or 50%

Adding Weights

Sample weights must be used when tabulating MICS data unless your country used a self-weighting sample. There are three kinds of weights: household, women's and children's. Household weights should be used with the household and household listing data files. Women's weights should be used with the women's data file. Children's weights should be used with the children's data file.

Weights must be used if the probability of selecting a household for interview was not equal across all of your country. For example, consider the imaginary country Popsylvania. In the North region, 5 households per 10,000 were selected for interview. In the capital, Vladisville, 28 households per 10,000 were selected for interview. In the country as a whole, 10 households per 10,000 were selected for interview.

Sample weights must be used to adjust the sample to produce accurate estimates for the whole country. The sample weights used are the inverse of the relative probabilities of selection. For example, the sample weights for the North region and Vladisville are

North Region	10/5	2.00
Vladisville	10/28	0.35

The weights for your country are calculated in a spreadsheet named WEIGHTS.XLS. The first step is to run the SPSS program WGTTAB.SPS. This program will produce the information needed to complete the WEIGHTS.XLS spreadsheet. Add this information to WEIGHTS.XLS and it will automatically calculate sample weights in columns Q, W and AC.

Save WEIGHTS.XLS when you have made these changes. Then close Excel and open SPSS. From the file menu, select the open option. Select WEIGHTS.XLS and click the open button. SPSS will pop-up the opening files options dialog box. Enter in this box the range of cells in the spreadsheet that you want to read. In the standard version, the range of cells is A5:AD16.

When you have entered the range of cells, click ok. SPSS will read in the specified range of cells and assign variable names to each column based on the values in row 5. In addition to the rows of the spreadsheet that you need, SPSS will also read in three rows that you do not need. Delete rows 1, 6 and 7 from your SPSS data set. Save the file in the SPSS sub-directory under the name WEIGHTS.SAV when these modifications have been made.

WEIGHTS.SPS merges the appropriate weights in WEIGHTS.SAV onto HH.SAV, HL.SAV, WM.SAV and CH.SAV using a macro. WEIGHTS.SPS also deletes unnecessary variables in WEIGHTS.SAV before it merges it with the data files. WEIGHTS.SPS does not need to be modified if your sample is stratified based on area and region.

Output

When labels, variables and weights have been added to these SPSS files and variables have been recoded, you will have four analysis files. The tabulation programs will use these four data files to create the tables in the tabulation plan.

Tabulation

There is one tabulation program for each table. 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.

Each tabulation program must be carefully reviewed. One important thing to check is 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 reference 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 included in tabulations.

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 1 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.

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 4 contains information on the percent 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 weight percent 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.

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.

The Include Command

The SPSS program TABLES.SPS can be used to run all of the tabulation programs at once. It consists of a series of INCLUDE commands that execute the tabulation programs individually. If

SPSS encounters an error in a program that is included (i.e., executed by an INCLUDE command), it will immediately stop executing the program and return to the program that included the program (i.e., the program that contained the INCLUDE command).

Because of this, you should only use TABLES.SPS when you have checked and modified all of the individual tabulation programs. Be sure also to 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 indented lines can be preceded by the ‘+’ character. The commands below, from the program T16.SPS, illustrate the use of the ‘+’ character to maintain program structure.

```
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. Consider the commands below from the program T1.SPS.

```
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 MICS programming style.)

The third and fourth restrictions 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 for each of the three data entry packages. The programs are named EPIINFO.SPS, IMPS.SPS and ISSA.SPS. These programs should only be used when all of the component programs have been executed and shown to work.

These programs are useful for recreating analysis files when a change is made to one of the file creation programs. They insure that all of the programs will be executed and that they will be executed in the proper order.

Appendix A – Files

File type	Questionnaire	Filename	Subdirectory
Data description			
	Household	MICSHH.QES	ENTRY
	Household listing	MICSHL.QES	ENTRY
	Women	MICSWM.QES	ENTRY
	Children	MICSCH.QES	ENTRY
Data files[1]			
	Household	HH###&.REC	DATA
	Household listing	HL###&.REC	DATA
	Women	WM###&.REC	DATA
	Children	CH###&.REC	DATA
Data entry applications			
	Household	MICSHH.CHK	ENTRY
	Household listing	MICSHL.CHK	ENTRY
	Women	MICSWM.CHK	ENTRY
	Children	MICSCH.CHK	ENTRY
Structure check			
	All	CHECK.PGM	SUPER
Secondary editing			
	All	EDITING.PGM	SUPER
Exporting data			
	Household	MERGEHH.BAT	EXPORT
	Household listing	MERGEHL.BAT	EXPORT
	Women	MERGEWM.BAT	EXPORT
	Children	MERGECH.BAT	EXPORT
Reading data into SPSS			
	Household	READHH.SPS	SPSS
	Household listing	READHL.SPS	SPSS
	Women	READWM.SPS	SPSS
	Children	READCH.SPS	SPSS
Variable and value labels			
	Household	LABELHH.SPS	SPSS
	Household listing	LABELHL.SPS	SPSS
	Women	LABELWM.SPS	SPSS
	Children	LABELCH.SPS	SPSS
Analysis file creation			
	Household listing	MAKEHL.SPS	SPSS
	Women	MAKEWM.SPS	SPSS
	Children	MAKECH.SPS	SPSS
Adding weights			
	All	WEIGHTS.SPS	WEIGHTS
Tabulation[2]			
	All	T??.SPS	SPSS

1. The # characters in the data file names represent the cluster number; the & character is either an M (main) or a V (verification).
2. The ? character in the tabulation program names represents the table name.