**DATA PROCESSING PROCEDURES 02:**

**PREPARING THE CAPI DATA ENTRY SYSTEM**

Table of Contents

[1. Introduction to the .IN File 4](#_Toc52619876)

[1.1 Steps to Create the .IN File 4](#_Toc52619877)

[1.2 Steps to Modify the .IN File 8](#_Toc52619878)

[1.3 An Example from Rwanda 9](#_Toc52619879)

[1.4 The .IN File for DHS and MIS Surveys 10](#_Toc52619880)

[2. Modifying your DHS Country Dictionary 17](#_Toc52619881)

[3. Modifying your DHS Country Forms 23](#_Toc52619882)

[4. Adjusting the Logic for the CAPI Data Entry Programs 31](#_Toc52619883)

[4.1 What is New in CollectHH Application Logic 31](#_Toc52619884)

[4.1.1 The PROC GLOBAL 31](#_Toc52619885)

[4.1.2 GPS Collection in the Household Data Capture Application 34](#_Toc52619886)

[4.1.3 Warning Instead of Errmsg 34](#_Toc52619887)

[4.1.4 Function tr 34](#_Toc52619888)

[4.1.5 Messages in Multiple Languages (in .MGF file) 35](#_Toc52619889)

[4.1.6 Blocks 35](#_Toc52619890)

[4.1.7 The DHS Program Developed Functions That Are Now Part of CSPro 36](#_Toc52619891)

[4.1.8 Recode/Endrecode Statement Instead of Box/Endbox Statement 36](#_Toc52619892)

[4.1.9 New variables added as part of DHS-8 Questionnaire Revision 37](#_Toc52619893)

[4.2 Good Programming Practices 38](#_Toc52619894)

[4.2.1 Indenting the Code 38](#_Toc52619895)

[4.2.2 Documenting the Program by Adding Comments 39](#_Toc52619896)

[4.2.3 Other Good Practices 39](#_Toc52619897)

[4.3. Adjusting the Logic for the Data Entry Programs CollectHH & CollectIN 39](#_Toc52619898)

[4.3.1 Run A2Q on Application Files 40](#_Toc52619899)

[4.3.2 Household Application – COLLECTHH 42](#_Toc52619900)

[4.4 Biomarker Questionnaires 54](#_Toc52619901)

[4.4.1 Checking Second Decimal by Type of Scale 54](#_Toc52619902)

[4.4.2 Checking Height is Bigger than Weight 54](#_Toc52619903)

[4.4.3 Double Entering Measurements for Weight, Height, and Hemoglobin 55](#_Toc52619904)

[4.5 QSF 55](#_Toc52619905)

[4.5.1 QSF Editor 58](#_Toc52619906)

[4.5.2 Questionnaire Extraction Tools 58](#_Toc52619907)

[4.6 Options Menu 58](#_Toc52619908)

[4.6.1 Data Entry Option 58](#_Toc52619909)

[4.6.2 Data Sources Option 59](#_Toc52619910)

[4.6.3 Paradata Options 60](#_Toc52619911)

[4.7 Testing your Programs 60](#_Toc52619912)

# 1. Introduction to the .IN File

All DHS surveys are created using Core Standard survey templates, utilities, and applications. This includes questionnaires, data dictionaries, data processing programs, even documentation. Typically, though, nearly all countries request the addition of new Country Specific questions and the deletion of Core Standard questions. This can apply to entire sections or modules too!  A country may also want to obtain more detailed information on a specific subject. Not only are there Country Specific questions, there are also Country Specific answers.  Core Standard questions can have Country Specific answers added or several Core Standard answers can be aggregated into a single Country Specific answer.  Occasionally, responses to questions having “Other” answers are recoded as new Country Specific answers.

For efficiency, simplicity, and accuracy, a key, known as the .IN file is created to establish a link between the Core Standard and the Country Specific survey components. Every country differs, meaning every key will differ and need to be modified.  This task is done by the country's data processor specialist. The accuracy of the .IN file is critical. As noted earlier the .IN file is used multiple times to create a survey's various components so taking the extra time to create a high-quality Country Specific .IN file will save time in the long run.

Your task is to modify the Standard Core .IN file and create a key establishing links between the Core Standard survey questions and your country's survey questions. Please follow the steps to modify the .IN file to complete this task!

The time to prepare an .IN file can take several hours (approximately 1 full day) so be sure to allow yourself plenty of time to create .IN file that is both accurate and complete.

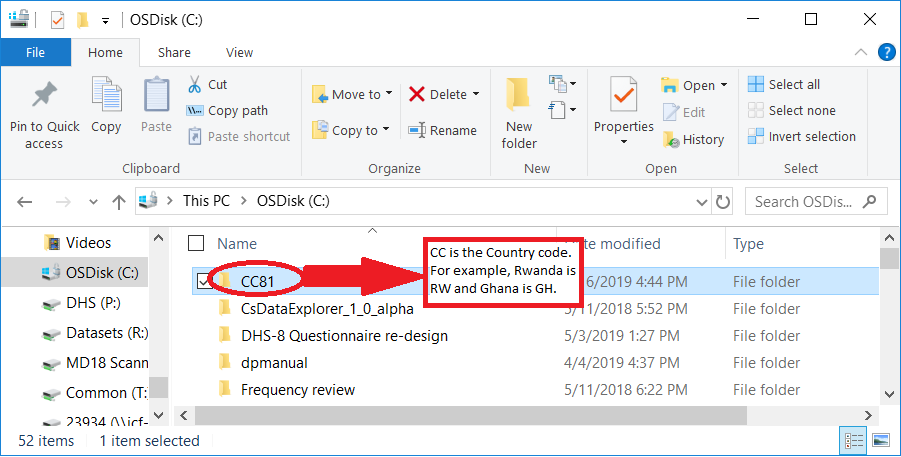
## 1.1 Steps to Create the .IN File

|  |  |
| --- | --- |
| **Country** | **CC** |
| Ethiopia | ET |
| Liberia | LB |
| Nigeria | NG |
| Pakistan | PK |
| Rwanda | RW |
| Uganda | UG |
| Zambia | ZM |

Here are the steps to create your .IN file:

**Step 1:** Create a folder on your C: drive named CC81, and replace CC by your two-letter country abbreviation. For example, Ghana would be GH. Then, copy folders \Dicts and \Entry from

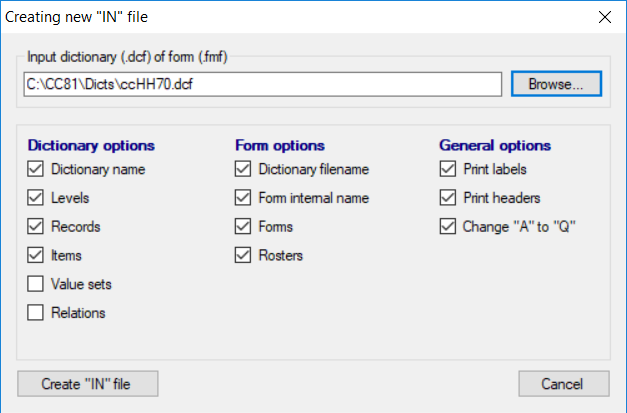
SharePoint\Standard Project\CC71 into C:\CC81.



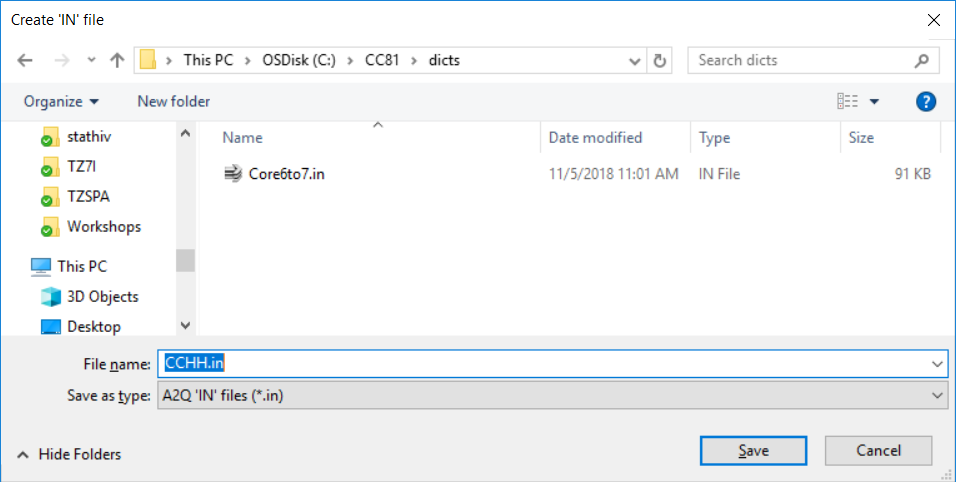
**Step 2:** Open A2Q to create the IN file by clicking the first icon from the left:



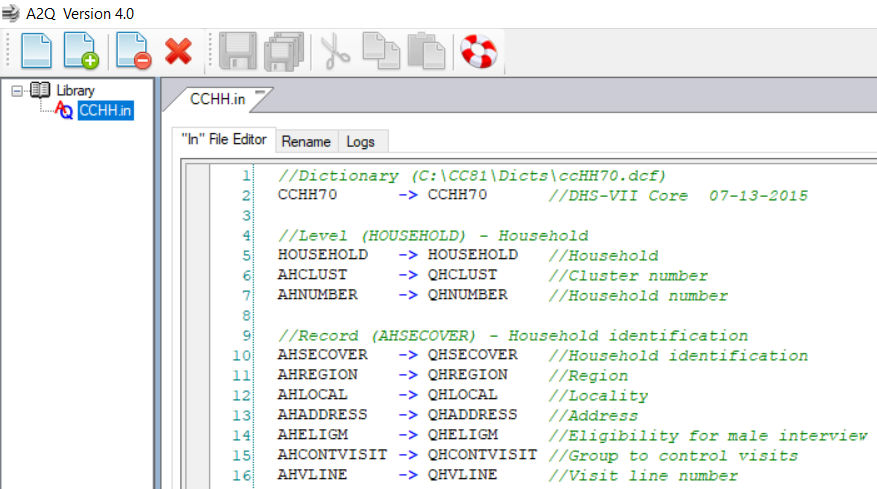
In the window that opens, **browse** to find the dictionary file for the Household Questionnaire located in C:\CC81\Dicts\ccHH70.dcf and click Open.



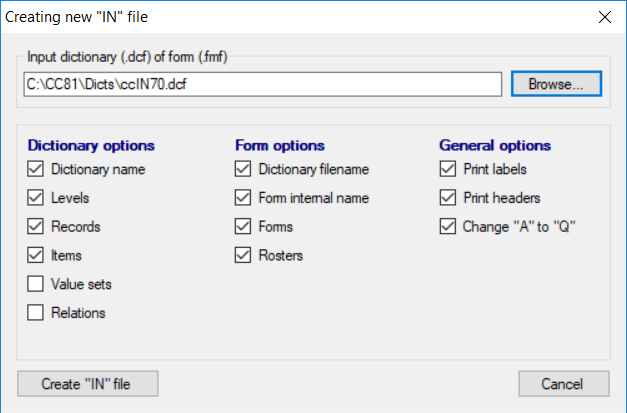
Click on “Create “IN” file”. Save your file under directory C:\CC81\Dicts, name your “IN” files as CCHH.in, replacing CC by your country code. Then, click Save.

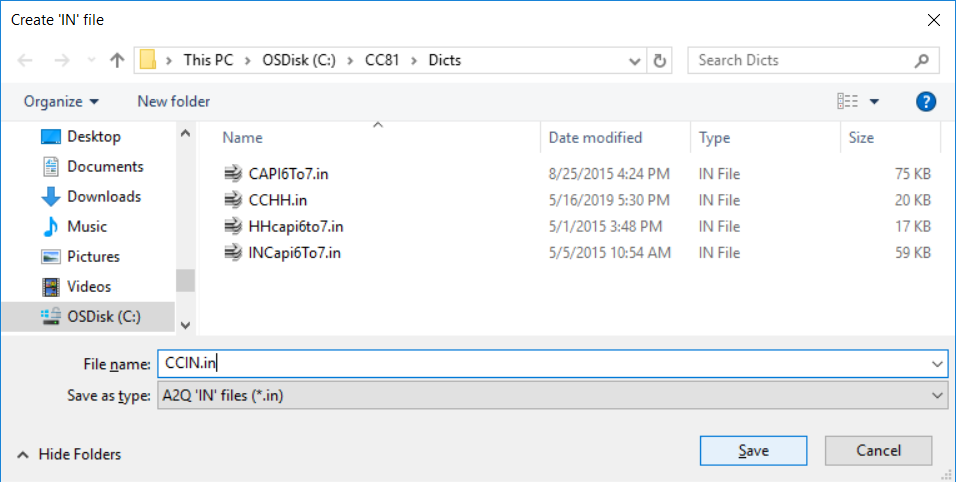


The following window will appear showing the IN file for the Household Questionnaire:

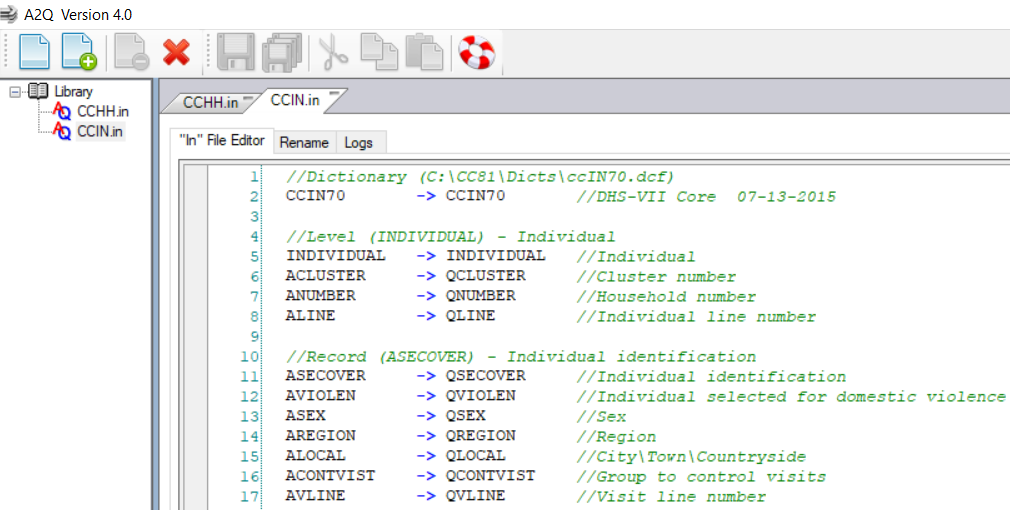


Do the same steps to create the “IN” file for the Individual Questionnaire, using dictionary ccIN70.dcf (see screenshots below).





There are two .IN files, one for the Household and one for the Individual Questionnaires opened in the A2Q editor. Now you are ready to begin editing the .IN files!



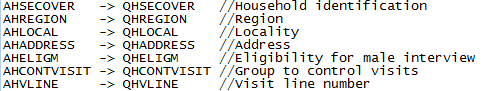
## 1.2 Steps to Modify the .IN File

**Step 3:** Begin working with the IN file for the Household Questionnaire. Open both, the Core Standard Questionnaires located in SharePoint\Standard Project\Standard DHS – Data Collection Library\Model Questionnaires\English (for DHS surveys) or SharePoint\Standard Project\Standard MIS – Data Collection Library\Questionnaires\English (for MIS surveys), and your Country questionnaires provided by the Survey Manager.

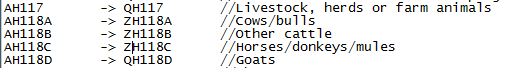


**Step 4:** For each difference between the Core Standard QREs and your Country QREs, modify the .IN file. Use the following tips and rules:

* To the left of the "->” are the names of the Core Standard variables (nearly always beginning with the letter A). To the right of the "->” are the names of your Country variable. Note that the letter Q has replaced the first letter A. Also note that in many, if not most cases, that is the only difference. These are the Core Standard variables. After the "//”, are the labels for each variable.



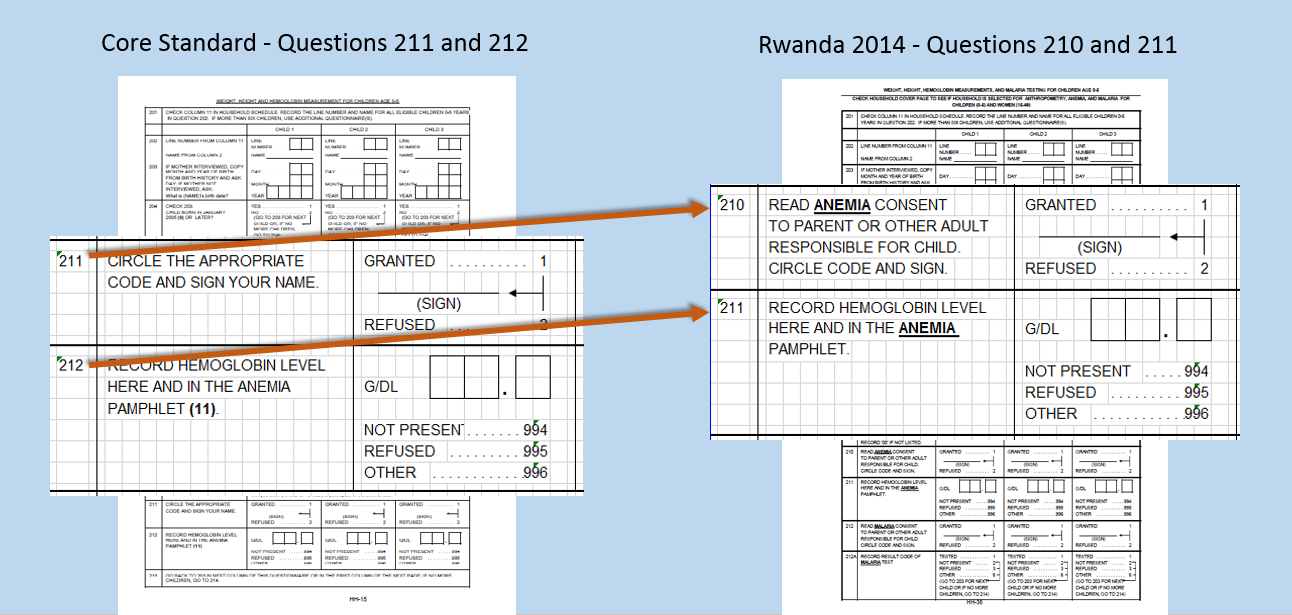
* In this case the .IN file for your Country questionnaire is exactly like the Core Standard. Again, every variable on the left is exactly the same as on the right, except A is replaced by Q. However, no country's questionnaire matches the Core Standard. Because of this, the right side variables need to be modified to match your Country questionnaires.
* NEVER change the name of the variable on the left side. All variables on the left begin with an "A" and need to remain that way to correspond to the Core Standard questionnaire.
* Carefully pay attention to the variable labels on the far right, appearing after the "//”. These labels are used to identify the questions in your Country questionnaire that match those in the Core Standard questionnaire.
* For this task focus only on the questions and not the response categories. Differences in response categories are not reflected in the .IN file. Response categories will be addressed later when working with the dictionary.
* If the variable has been DROPPED from your Country questionnaires, replace the "Q" with a "Z" or “ZZZ” (for more visibility) to indicate it has been dropped and is not included in your Country questionnaire. Below is an example where the variables for cows/bulls, other cattle, and horses/donkeys/mules have been dropped from the questionnaires and a "Z" has replaced the "Q".

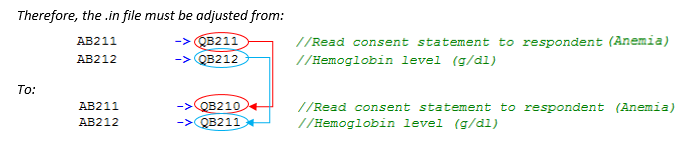


* For questions added to your Country questionnaire, which are called country specific questions, do NOTHING to the .IN file. Although these questions will be included in the programs, they are not included in the .IN file because they have no corresponding questions in the Core Standard to do the replacement. However, highlight with a colored marker the new question in your Country questionnaire to denote this is a Country Specific question you will need to add to the dictionary at a later stage.
* For questions that have been re-ordered or re-numbered in your Country questionnaire in comparison to the Core Standard questionnaire, change the number of the Q variable on the right to match your Country questionnaire. As mentioned earlier, never change the variable name on the left. See the example below from a survey in Rwanda.

## 1.3 An Example from Rwanda

In Rwanda 2014 DHS, questions in the biomarker section were numbered slightly differently from those in the core. Below on the left side are the Core Standard questions on consent and hemoglobin levels. These are numbered 211 and 212, respectively. However, in the Rwanda survey, the same consent and hemoglobin levels questions were numbered as 210 and 211.





**Step 5:** There are a number of variables in the Core Standard questionnaire that you will not find in your Country questionnaire after carefully comparing and reading the variable labels (those appearing after the “//”). When searching, you will not find these variables, not because they were dropped from your Country questionnaires, but because these are **calculated variables** that will be populated at a later stage or they are **auxiliary variables** that helps controlling the CAPI data collection application. Thus, you should not add “Z” or “ZZZ” to these variables; they should remain in your dictionary file **as long as they are applicable or the section where they belong (or are located) is present in your Country questionnaire**. As you become familiar in your role as Data Processor for the DHS, you will get to know these variables, meanwhile below is a reference list of these variables by survey type (DHS or MIS).

## 1.4 The .IN File for DHS and MIS Surveys

**FOR DEMOGRAPHIC AND HEALTH SURVEYS (DHS)**

**Household Questionnaire**

**QHSECOVER - Household identification**

AHINTC -> QHINTC //Date of household interview (CMC)

AHINTCD -> QHINTCD //Date of household interview Century Day Code (CDC)

AHWEIGHT -> QHWEIGHT //Household weight (6 decimals)

AMWEIGHT -> QMWEIGHT //Household weight - men's subsample (6 decimals)

AHSTRATA -> QHSTRATA //Sampling errors stratum

AHWLTHF -> QHWLTHF //Wealth index factor (5 decimals)

AHWLTHI -> QHWLTHI //Wealth index quintile

AHTYPE -> QHTYPE //Type of place of residence

AHWLTFUR -> QHWLTFUR //Wealth index factor for urban/rural (5 decimals)

AHWLTIUR -> QHWLTIUR //Wealth index quintile for urban/rural

AHALTIT -> QHALTIT //Cluster altitude in meters

AHFINISH -> QHFINISH //Handler to end interview

**QHSEC01 - Household Schedule**

ML07I -> ML07I //Age corrected from individual questionnaire

ML07F -> ML07F //Flag for age correction

MLPREG -> MLPREG //Pregnancy status from individual questionnaire

**QBSECOVER - Biomarker Identification**

ABINTC -> QBINTC //Date of household interview (CMC)

ABINTCD -> QBINTCD //Date of biomarker questionnaire Century Day Code (CDC)

**QBSEC01 - Height, Weight, Hemoglobin (children)**

ACCMC -> QCCMC //Child's CMC of birth

ACCDC -> QCCDC //Child's Century Day Code (CDC) of birth

ACFLG -> QCFLG //Date flag for child's date of birth

ACDBR -> QCDBR //Imputed day of birth from birth history

ACMBR -> QCMBR //Imputed month of birth

ACYBR -> QCYBR //Child's year of birth

ACPC1 -> QCPC1 //Ht/A Percentile

ACSD1 -> QCSD1 //Ht/A Standard deviations

ACRM1 -> QCRM1 //Ht/A Percent of ref. median

ACPC2 -> QCPC2 //Wt/A Percentile

ACSD2 -> QCSD2 //Wt/A Standard deviations

ACRM2 -> QCRM2 //Wt/A Percent of ref. median

ACPC3 -> QCPC3 //Wt/Ht Percentile

ACSD3 -> QCSD3 //Wt/Ht Standard deviations

ACRM3 -> QCRM3 //Wt/Ht Percent of ref. median

ACMLN -> QCMLN //Mother's line number from individual questionnaire

ACBO -> QCBO //Birth order of child

ACBI -> QCBI //Birth interval

ACED1 -> QCED1 //Mother's highest educational level

ACED2 -> QCED2 //Mother's highest grade at that level

ACED3 -> QCED3 //Mother's highest educational level (CS prelim&final report)

ACSD4 -> QCSD4 //Ht/A Standard deviations (according to WHO)

ACSD5 -> QCSD5 //Wt/A Standard deviations (according to WHO)

ACSD6 -> QCSD6 //Wt/Ht Standard deviations (according to WHO)

ACSD7 -> QCSD7 //BMI Standard deviations (according to WHO)

ACNAME -> QCNAME //Child name

ACAUXIL -> QCAUXIL //Auxiliar to decide if Biomarkers for children are recorded at

this time

**QBSEC02 - Height, Weight, Hemoglobin (women)**

AWCMC -> QWCMC //CMC of birth of woman

AWFLG -> QWFLG //Date flag for woman's date of birth

AWMBR -> QWMBR //Imputed month of birth

AWYBR -> QWYBR //Woman's year of birth

AWSMK -> QWSMK //Respondent smoke practices

AWPRG -> QWPRG //Respondent pregnancy status

AWMOS -> QWMOS //Months of pregnancy

AWPC1 -> QWPC1 //Ht/A Percentile (resp.)

AWSD1 -> QWSD1 //Ht/A Standard deviations (resp

AWRM1 -> QWRM1 //Ht/A Percent ref. median (resp

AWRM3 -> QWRM3 //Wt/Ht Percent ref. median (DHS

AWRM4 -> QWRM4 //Wt/Ht Percent ref. median (Fog

AWRM5 -> QWRM5 //Wt/Ht Percent ref. median (WHO

AWSD3 -> QWSD3 //Wt/Ht Std deviations(resp) DHS

AWBM -> QWBM //Body mass index for respondent

AWRI -> QWRI //Rohrer's index for respondent

AWRESI -> QWRESI //Result of individual interview

AWED1 -> QWED1 //Woman's highest educational level

AWED2 -> QWED2 //Woman's highest year of education

AWED3 -> QWED3 //Highest educational level (CS for preliminary & final report)

AWHIVWGT -> QWHIVWGT //HIV weight (6 decimals)

AWNAME -> QWNAME //Woman name

AWAUXIL -> QWAUXIL //Auxiliar to decide if Biomarkers for women are recorded at

this time

**QBSEC03 - Height, Weight, Hemoglobin (men)**

AMCMC -> QMCMC //CMC of birth of man

AMFLG -> QMFLG //Date flag for man's date of birth

AMMBR -> QMMBR //Imputed month of birth

AMYBR -> QMYBR //Man's year of birth

AMSMK -> QMSMK //Respondent smoke practices

AMPC1 -> QMPC1 //Ht/A Percentile (resp.)

AMSD1 -> QMSD1 //Ht/A Standard deviations (resp)

AMRM1 -> QMRM1 //Ht/A Percent ref. median (resp)

AMRM3 -> QMRM3 //Wt/Ht Percent ref. median (DHS)

AMRM4 -> QMRM4 //Wt/Ht Percent ref. median (Fog)

AMRM5 -> QMRM5 //Wt/Ht Percent ref. median (WHO)

AMSD3 -> QMSD3 //Wt/Ht Std deviations(resp) DHS

AMBM -> QMBM //Body mass index for respondent

AMRI -> QMRI //Rohrer's index for respondent

AMRESI -> QMRESI //Result of individual interview

AMED1 -> QMED1 //Man's highest educational level

AMED2 -> QMED2 //Man's highest year of education

AMED3 -> QMED3 //Highest educational level (CS for preliminary & final report)

AMHIVWGT -> QMHIVWGT //HIV weight (6 decimals)

AMNAME -> QMNAME //Man name

AMAUXIL -> QMAUXIL //Auxiliar to decide if Biomarkers for men are recorded at this

time

**Individual Questionnaire**

**QSECOVER - Individual identification**

ASEX -> QSEX //Sex

AINTC -> QINTC //Date of interview (CMC)

AINTCD -> QINTCD //Date of interview Century Day Code (CDC)

AWEIGHT -> QWEIGHT //Sample weight (6 decimals)

ATYPE -> QTYPE //Type of place of residence

AQTYPE -> QQTYPE //Type of questionnaire

ADWEIGHT -> QDWEIGHT //Domestic violence weight (6 decimals)

ANAME -> QNAME //Name

ACORRECT -> QCORRECT //Is the correct individual

ACONSENT -> QCONSENT //Consent agreement

ABAUX1 -> QBAUX1 //Auxiliary to display all listed children

ABAUX2 -> QBAUX2 //Auxiliary to confirm that all children are listed

AINTROUSE -> QINTROUSE //Auxiliary to display introduction to contraceptive history

AFINAL -> QFINAL //Auxiliary to end woman's interview

AMFINAL -> QMFINAL //Auxiliary to end man's interview

**QWSEC01 - Respondent's Background**

A105C -> Q105C //Date of birth (CMC)

A105F -> Q105F //Date flag for A105C

A106C -> Q106C //Age of respondent calculated

A108A -> Q108A //Highest educational level

AWFTOTAL -> QWFTOTAL //All woman factor - total

AWFTYPE -> QWFTYPE //All woman factor - urban/rural

AWFREG -> QWFREG //All woman factor - regional

AWFEDUC -> QWFEDUC //All woman factor - educational

AWFWLTH -> QWFWLTH //All woman factor - wealth index

**QWSEC2B - Birth History**

ALAST5 -> QLAST5 //Child born in the last 5 years

A215C -> Q215C //Date of birth of child (CMC)

A215CD -> Q215CD //Date of birth of child Century Day Code (CDC)

A215DI -> Q215DI //Imputed day of birth

A215F -> Q215F //Date flag for A215C

A220C -> Q220C //Age at death months (imputed)

A220F -> Q220F //Date flag for A220C

**QWSEC2D - Reproduction (continued)**

A227C -> Q227C //Date of conception (CMC)

A227F -> Q227F //Date flag for A227C

A239F -> Q239F //Flag for A239

A224A -> Q224A //Births since January 201? (for immunizations)

A224B -> Q224B //Births since January 201? living with respondent (for food

consumption)

**QWSEC3A - Contraceptive Knowledge & Use**

A301OT -> Q301OT //Auxiliary to capture other methods

AALLMETH -> QALLMETH //Auxiliary to capture spontaneous knowledge of methods

**QWSEC3B - Contraceptive Practice**

A304N -> Q304N //Current contraceptive method (major method)

A309C -> Q309C //Date started using current method (CMC)

A309F -> Q309F //Date flag for A309C

**QWSEC3E - Contraceptive use history**

ACSOURCE -> QCSOURCE //Source of method

ACAUXM1 -> QCAUXM1 //Auxiliary month began using

ACAUXY1 -> QCAUXY1 //Auxiliary year began using

ACAUXM2 -> QCAUXM2 //Auxiliary month ended use

ACAUXY2 -> QCAUXY2 //Auxiliary year ended use

ACROWNEXT -> QCROWNEXT //Row where next episode start

ACDIRNEXT -> QCDIRNEXT //Direction for next episode

ACFEVENT -> QCFEVENT //End of interval between pregnancy-births-terminations

ACBEVENT -> QCBEVENT //Begin of interval between pregnancy-births-terminations

ACDURIN -> QCDURIN //Interval estimated before beginning of use

**QWSEC04) - Pregnancy and Postnatal Care**

A460F -> Q460F //Flag for amenorrhea

A463F -> Q463F //Flag for abstinence

**QWSEC07 - Marriage and Sexual Activity**

A710C -> Q710C //Date of first union (CMC)

A710F -> Q710F //Date flag for A710C

A711C -> Q711C //Age at first union (calculated)

A713F -> Q713F //Flag for A713

A714F -> Q714F //Flag for A714

**QWSEC09 - Husband's and Woman's Background**

A904A -> Q904A //Partners level of education

**QWSECM3 - Maternal Mortality**

AMM16C -> QMM16C //CMC date of birth of sibling

AMM17C -> QMM17C //CMC date of death of sibling

**QMSEC01 - [Men] Respondent's Background**

AM105C -> QM105C //Date of birth (CMC)

AM105F -> QM105F //Date flag for AM105C

AM106C -> QM106C //Age of respondent calculated

AM108A -> QM108A //Highest educational level

**QMSEC3A - [Men] Contraceptive Knowledge & Use**

AM301OT -> QM301OT //Auxiliary to capture other methods

AMALLMETH -> QMALLMETH //Auxiliary to capture spontaneous knowledge of methods

**QMSEC04 - [Men] Marriage and Sexual Activity**

AM411C -> QM411C //Date of first union (CMC)

AM411F -> QM411F //Date flag for AM411C

AM412C -> QM412C //Age at first union (calculated)

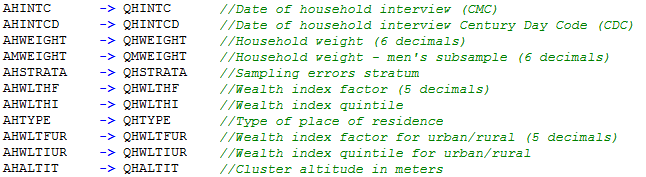
AM414F -> QM414F //Flag for AM414

AM415F -> QM415F //Flag for AM421

**FOR MALARIA INDICATOR SURVEYS (MIS)**

**Household Questionnaire**

**QHSECOVER**





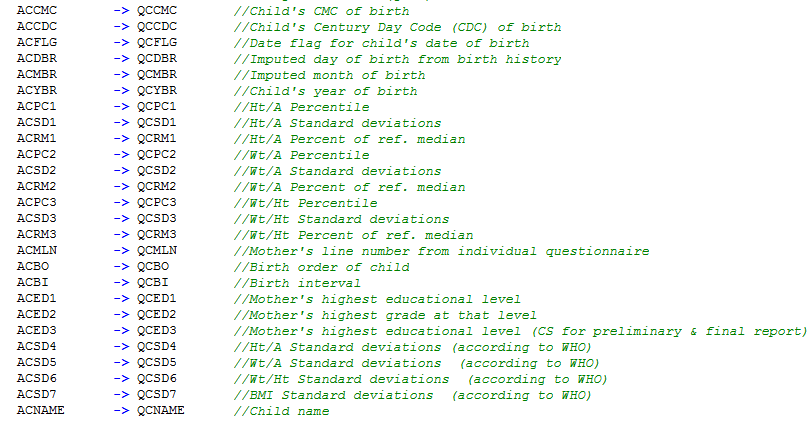
**QHSEC01**



**QBSECOVER**



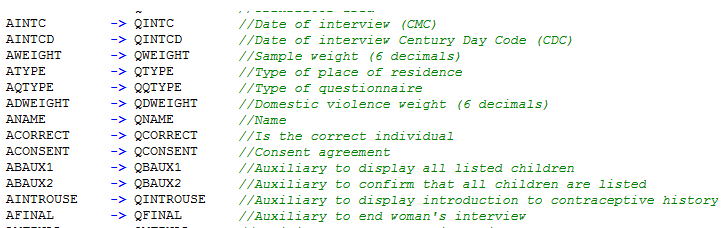
**QBSEC01**



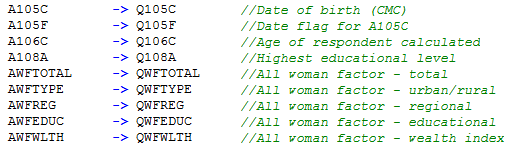
ACAUXIL -> QCAUXIL //Auxiliar to decide if Biomarkers for children are recorded at this time

**Woman Questionnaire**

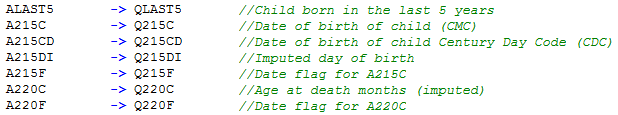
**QSECOVER**



**QWSEC01**



**QWSEC2B**



**QWSEC2D**



**Step 7:** Repeat the steps above for the Biomarker, Woman, and Man's questionnaires where applicable.

**Step 8:** Save your work OFTEN!

# 2. Modifying your DHS Country Dictionary

At this point you have finalized adjusting your .IN files after a thorough review of your Country questionnaire compared to the Core Standard questionnaire. The next step is to adjust your Country dictionary to add, modify, or delete variables or records according to DHS Conventions. Below are the steps to start modifying your DHS Country dictionary:

**Step 1:** Because we will be executing “Case Sensitive” replacements using A2Q and there are few references on the Form file that uses small letters instead of capitals, thus add another line at the top of both CCHH.in and CCIN.in as follows, where CC is your country code:

//Dictionary (C:\CC81\Dicts\ccHH70.dcf)

CCHH70 -> CCHH80 //DHS-VII Core 07-13-2015 //note: 80 because this is DHS-8 phase

ccHH70 -> CCHH80 //new line

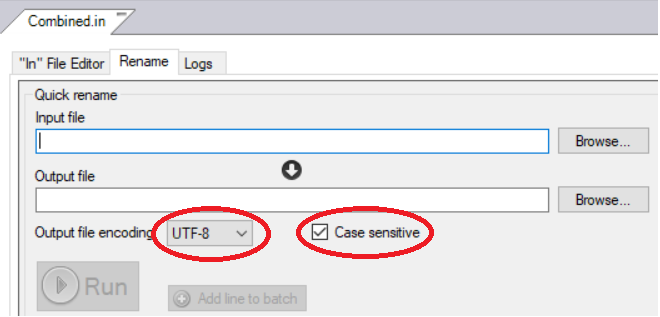
//Dictionary (C:\CC81\Dicts\ccIN70.dcf)

CCIN70 -> CCIN80 //DHS-VII Core 07-13-2015 //note: 80 because this is DHS-8 phase

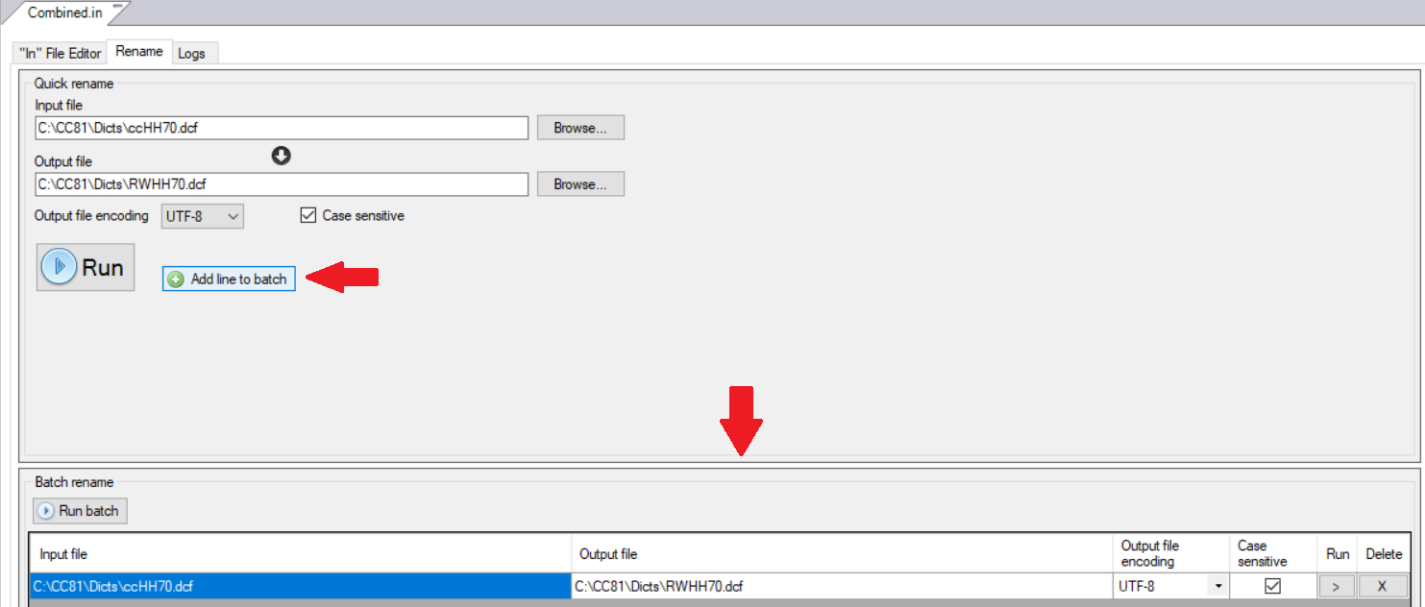
ccIN70 -> CCIN80 //new line

**Step 2:** Combine .IN files for CCHH.in and CCIN.in together using a text editor. Copy and paste the CCHH.in file located in your C:\CC81\Dicts folder and rename the resulting file as Combined.in. Open Combined.in using a text editor, go to the end, and click “Enter” to add a blank line. Open CCIN.in using a text editor and press “CTRL + A” to select all content, then copy and paste all the content at the end of the Combined.in file. Save your changes and close.

**Step 3:** Open the Combined.in file using A2Q and make replacements in batch using the Core Standard dictionaries and forms as input files. We will take advantage of the fact that A2Q manages multiple files at a time to make A-2-Q replacements at once for dictionaries and forms. Click on the “Rename” tab and make sure “file encoding” is UTF-8 and “Case sensitive” is ticked.



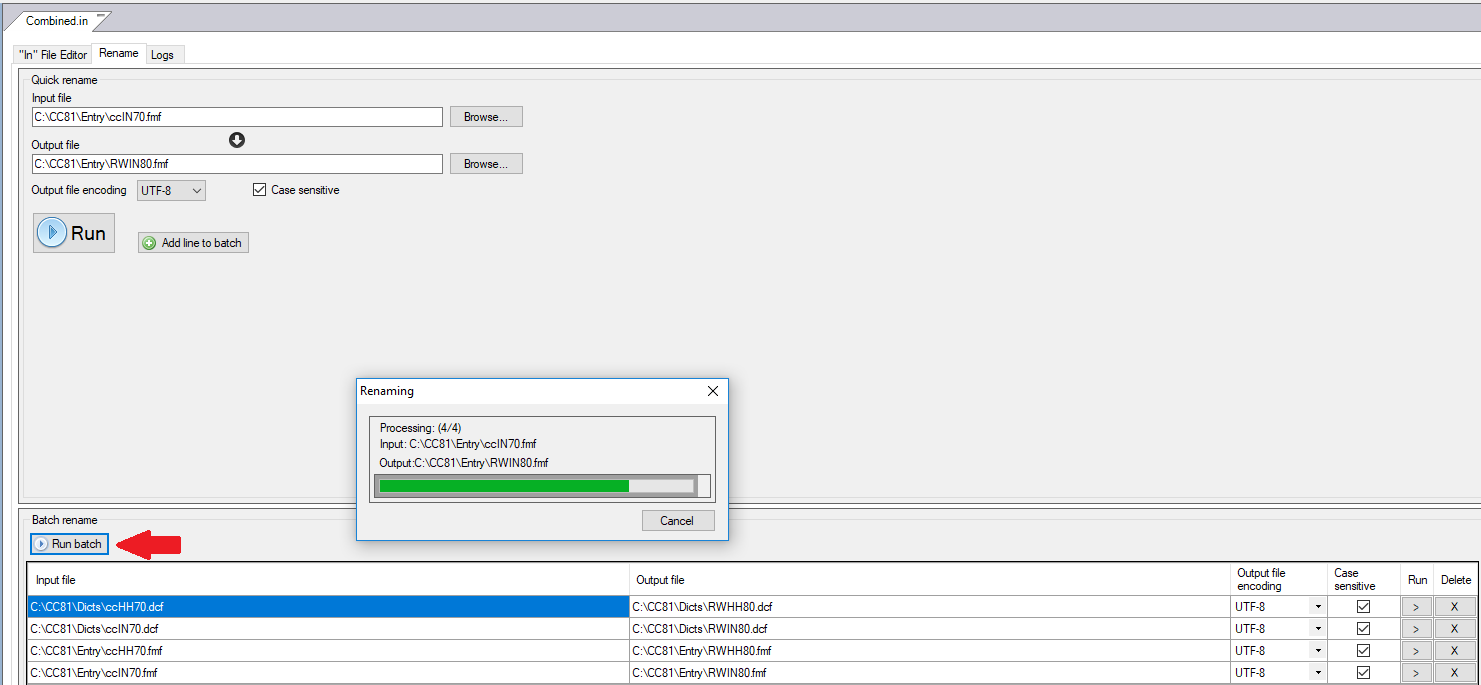
Browse for the **Input file** C:\CC81\Dicts\ccHH70.dcf. Copy and paste the filename and location from the Input file into the **Output file**, and rename the last to C:\CC81\Dicts\**cc**HH**8**0.dcf where cc is your country code and 8 represent DHS-8 current phase. Then click “Add line to batch” (see image below). You will notice that the line is placed at the bottom under the sub-section “Batch rename”.



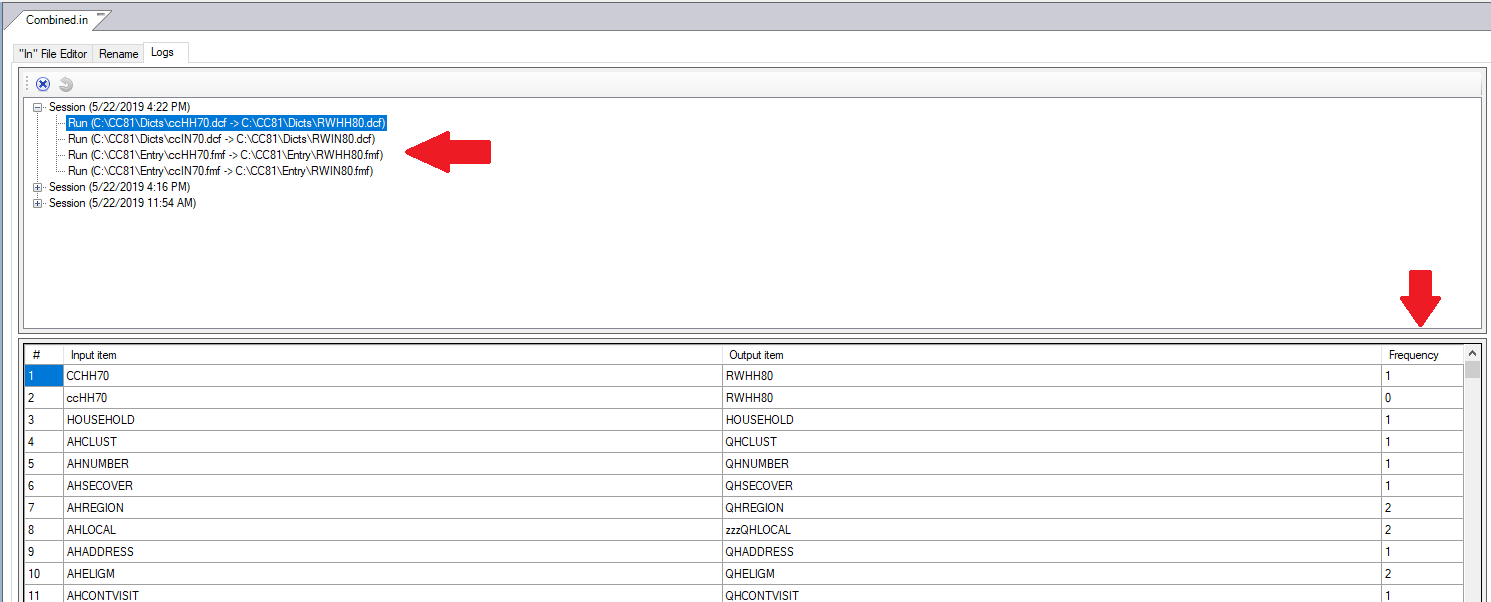
Repeat the above for the rest of the files (the Individual Core Standard dictionary and the Household and Individual Core Standard Forms):

|  |  |
| --- | --- |
| Input file | Output file |
| C:\CC81\Dicts\ccIN70.dcf | C:\CC81\**Dicts**\**cc**IN**8**0.dcf |
| C:\CC81\Entry\ccHH70.fmf | C:\CC81\**Entry**\**cc**HH**8**0.fmf |
| C:\CC81\Entry\ccIN70.fmf | C:\CC81\**Entry**\**cc**IN**8**0.fmf |

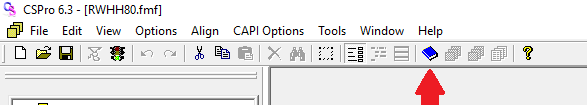
Then click on “Run batch”.



**Step 4:** When A2Q finalizes the “Run batch”, it moves to the “Logs” tab and shows the number of replacements done (under the Frequency column) for each of the variables and within each file. The image below is showing the replacements made to the first file C:\CC81\Dicts\RWHH80.dcf. To see the replacements made in other files, you can click on each file one by one.

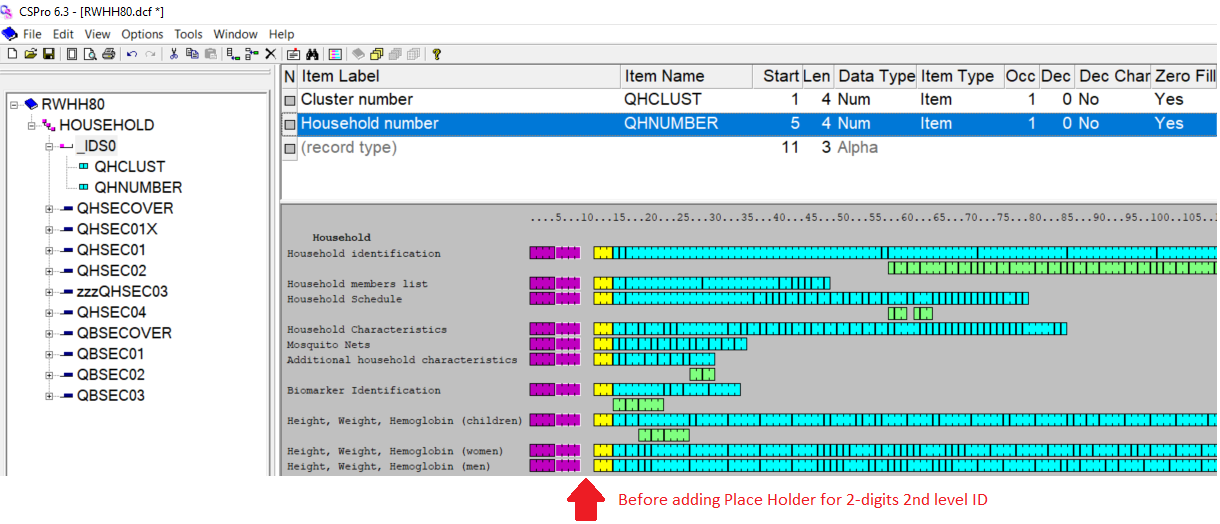


**Step 5:** Open the Household form file or .fmf for your country survey, in this example it is C:\CC81\**Entry**\**RW**HH**8**0.fmf. Opening the form file will also open the dictionary file associated to this form and allows to periodically refresh the contents on the forms when variables are deleted in the dictionary (i.e., when switching periodically between the dictionary view and the form view to allow the refresh on the form to happen). Then, switch to the dictionary view using the dictionary icon from the toolbar.

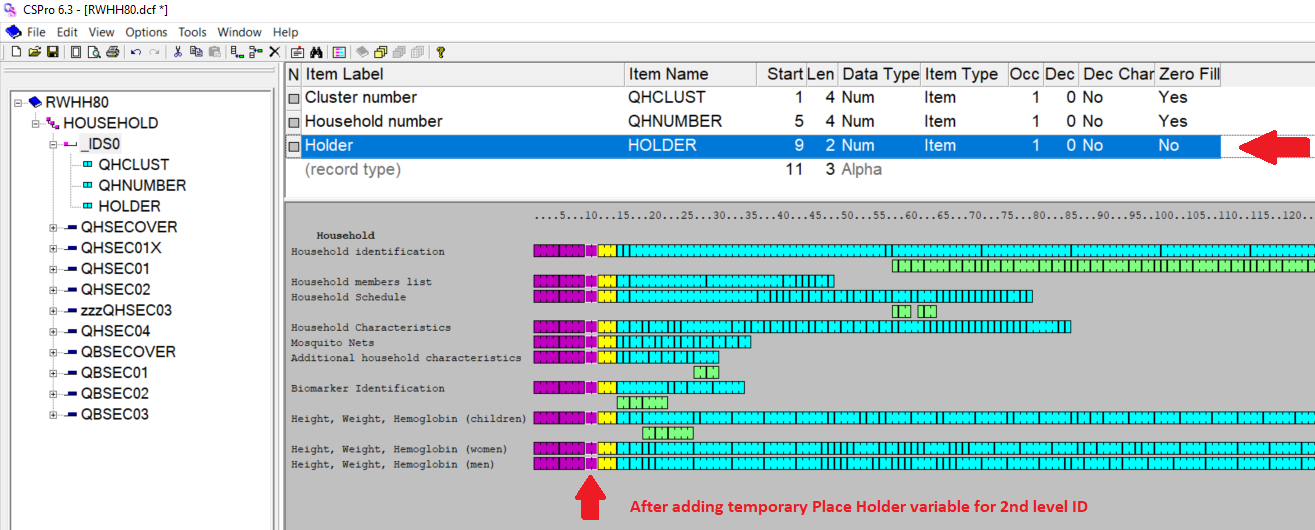


**Step 6:** The CAPI Household dictionary is set to Absolute position, meaning that inserting country specific variables in the proper place in a record would not be possible. Thus, if you have country specific questions that need to be added to the Household dictionary, you will need to turn Relative position on. Before doing such, you will need to temporarily add a 2-digits variable to the Household ID, that correspond to the 2-digits ID for the individual questionnaire. The reason is that CSPro manages hierarchical data and the Household and Individual data will be combined at a later stage, thus the 2-digits ID for the second level (Individual) is hold in the Household dictionary as blank with Absolute position. If you turn Relative position on without creating a place to hold the blanks, all records will shift backwards two digits and will not hold the blank space needed for the Individual ID. Therefore, temporarily add a “place holder” variable before changing the Options to Relative position that will allow you to insert country specific variables in the needed place.

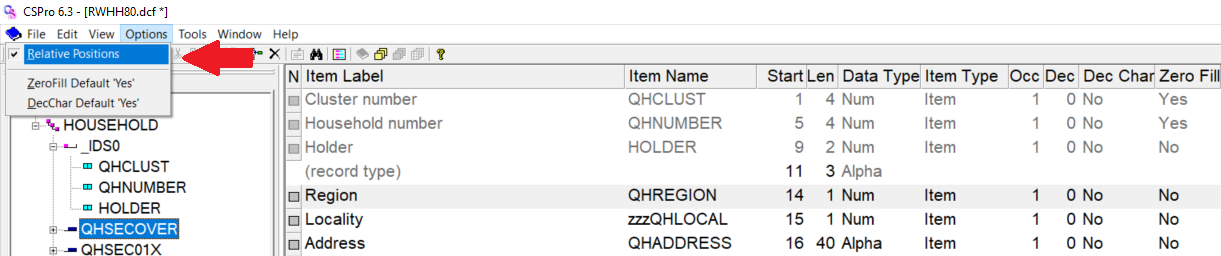
Before adding “place holder” variable:



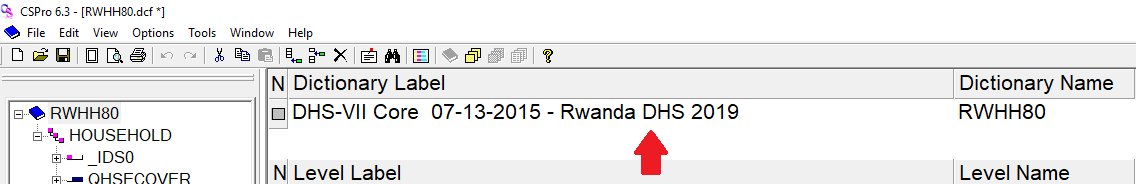
After adding temporary “place holder” variable:



Change CAPI Household dictionary to Relative position under “Options” menu, to allow inserting country specific variables to the proper place in a record.

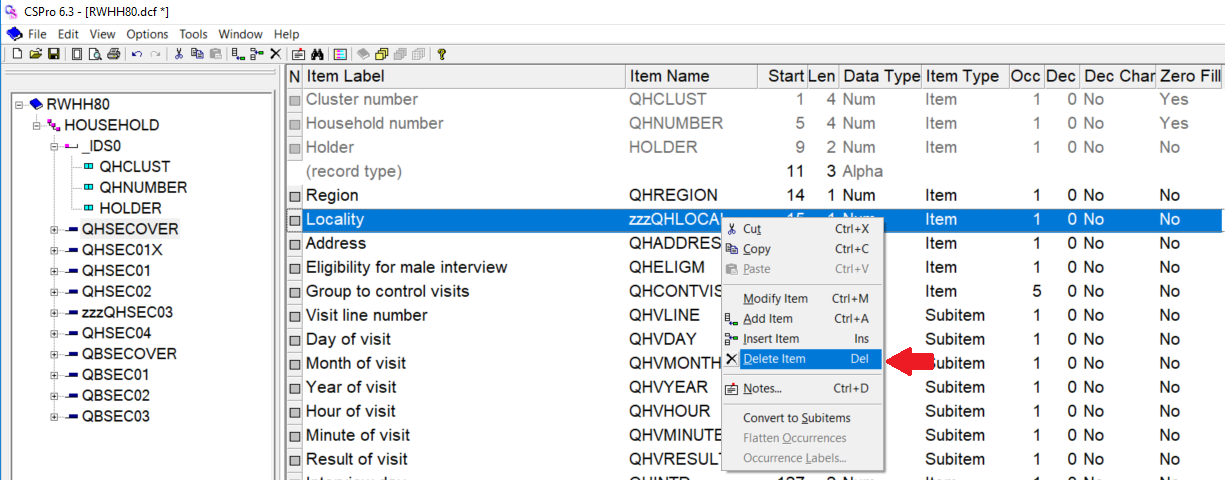


**Step 7:** Start modifying the dictionary from the very top of the Dictionary tree on the left panel. Add, modify, or delete variables or records from your country survey. While you modify the dictionary, save your changes OFTEN and do not forget to switch to the Form view periodically to allow the forms to refresh the changes made on the Household dictionary. Perhaps you can switch to the Form view after completing each record. The first change to be made into your country dictionary is adding your Country, Survey-type, and Year to the dictionary label, on our example it will be “ – Rwanda DHS 2019”.

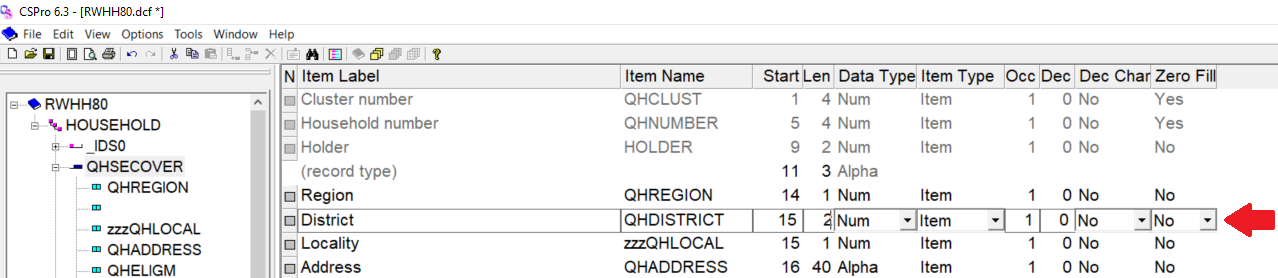


Then, review each record, variable by variable to:

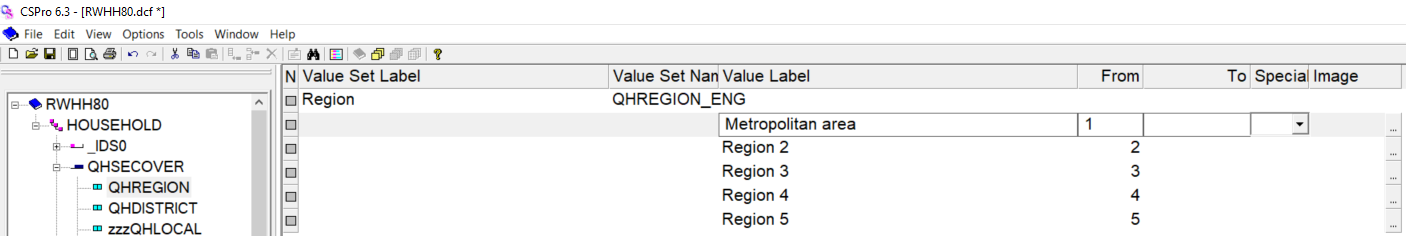
1. delete variables marked with zzz



2. add country specific variable in the proper place of a record (i.e., the place where it appears in the questionnaire)

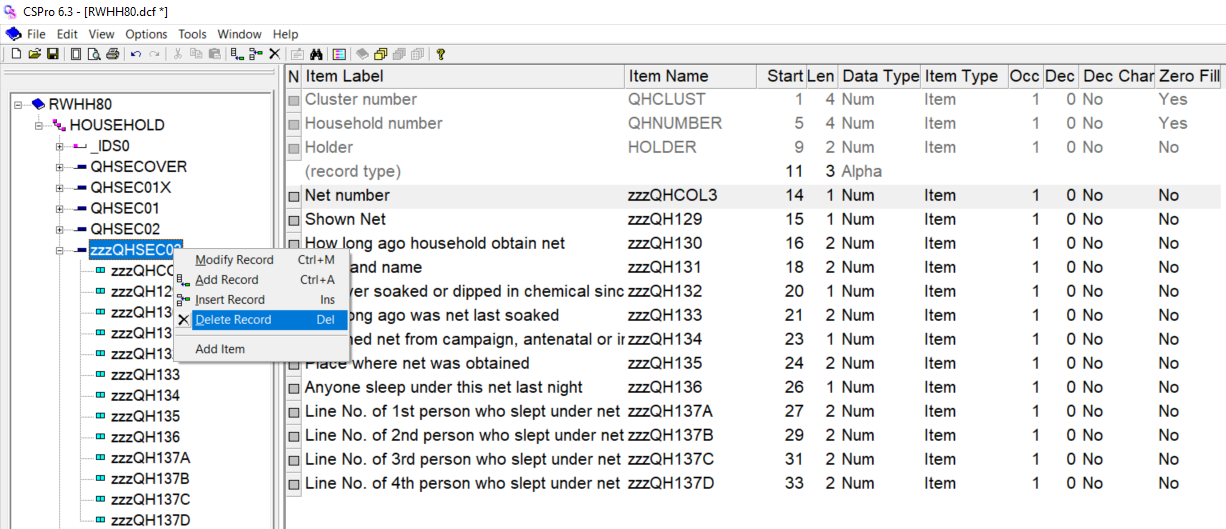


3. compare variable categories with the ones in your country survey and add/update any new category added or categories that have been modified. Also update the labels of standard variables with the proper country categories, for example, QHREGION. Label for Region in a country can be obtained from the Sample file. Ranges need to be updated as well, for instance, year of interview ranges from 2015-2020 in the standard Household dictionary, but since it is already year 2019, restrict the range to 2019-2020, if you expect the survey fieldwork will overlap for two years, or just 2019, if the survey fieldwork will end before December 31, 2019.

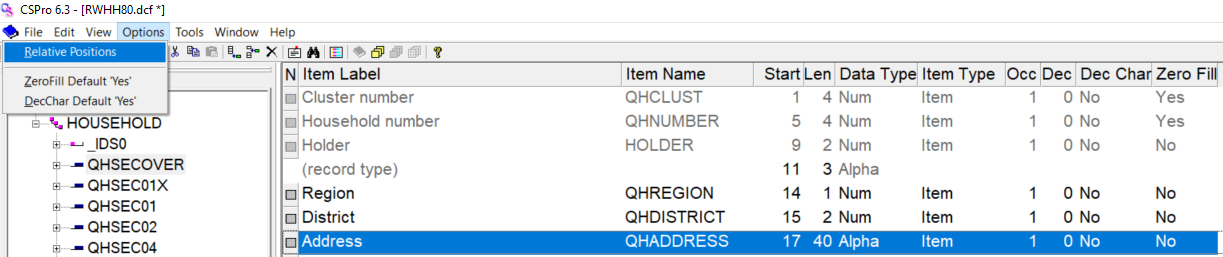


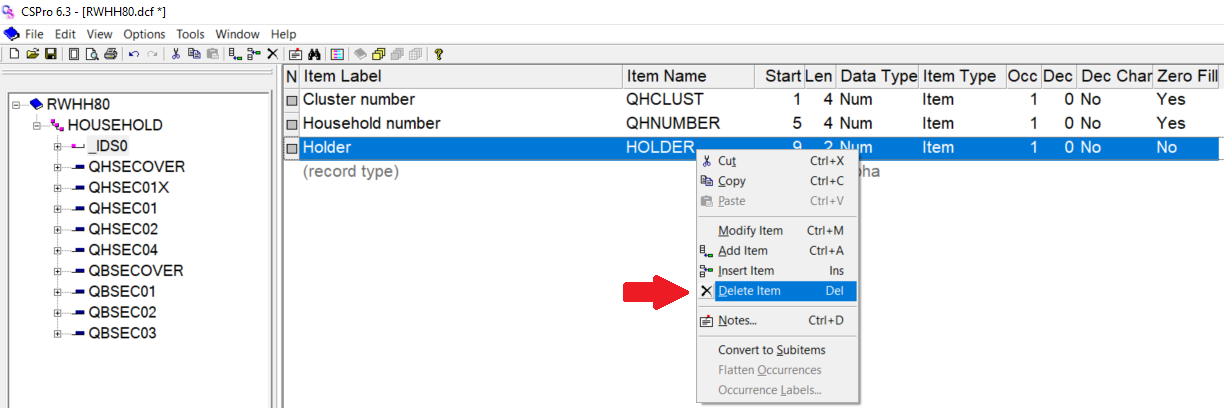


4. delete entire records when they have been marked with zzz or all variables inside the record are marked with zzz



**Step 8:** After you finish modifying the CAPI Household dictionary, turn the Relative position off under “Options” menu, and then delete the “place holder” variable from the ID.





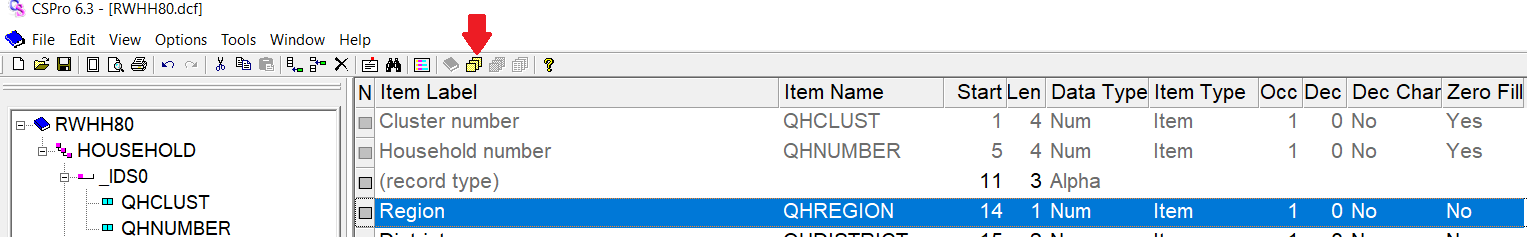
**Step 9:** Continue with your CAPI Individual Form and dictionary. Open the Individual form file or .fmf for your country survey, in this example it is C:\CC81\**Entry**\**RWIN8**0.fmf. Opening the form file will also open the dictionary file associated to it. Add, modify, or delete variables or records from your country survey and add/update variable categories. While you modify the Individual dictionary, save your changes OFTEN and do not forget to switch to the Form view periodically to allow the forms to refresh the changes made on the dictionary. Perhaps you can switch to the Form view after completing each record.

# 3. Modifying your DHS Country Forms

By now, you have completed your .IN files and dictionaries for your CAPI DHS country survey. Following instructions from your previous steps, the form files for the CAPI Household and Individual data collection already passed A2Q and all zzz marked variables have been refreshed/removed from the forms. Let us continue with the steps to further modify your form files.

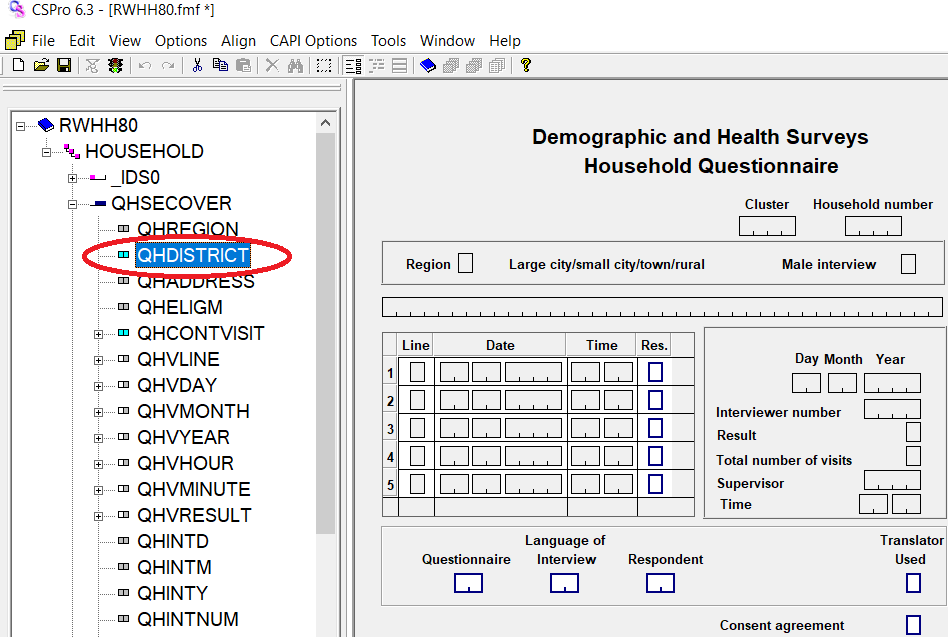
As a reminder, we are working with your DHS country Form files C:\CC81\**Entry**\**CC**HH**8**0.fmf and C:\CC81\**Entry**\**CC**IN**8**0.fmf. While modifying your Forms files, save your changes OFTEN.

**Step 1:** From the dictionary view, switch to the forms view by clicking the form icon on the tool bar. On the left panel, you will be switching between the “Dicts” tab and the “Forms” tab as you drag country specific variables from the dictionary into the proper form.

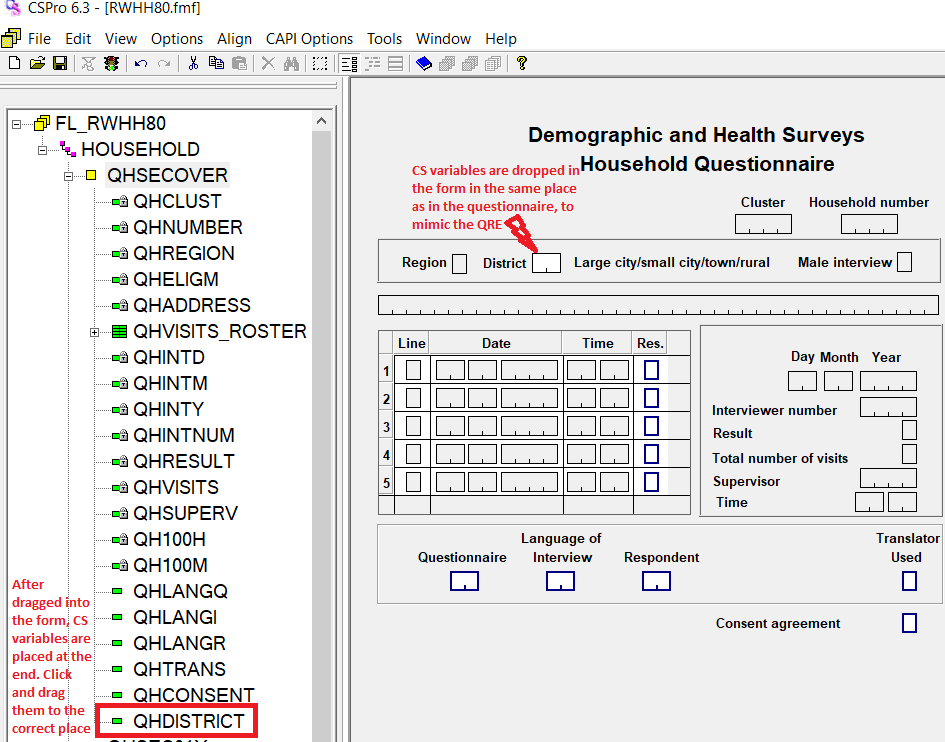




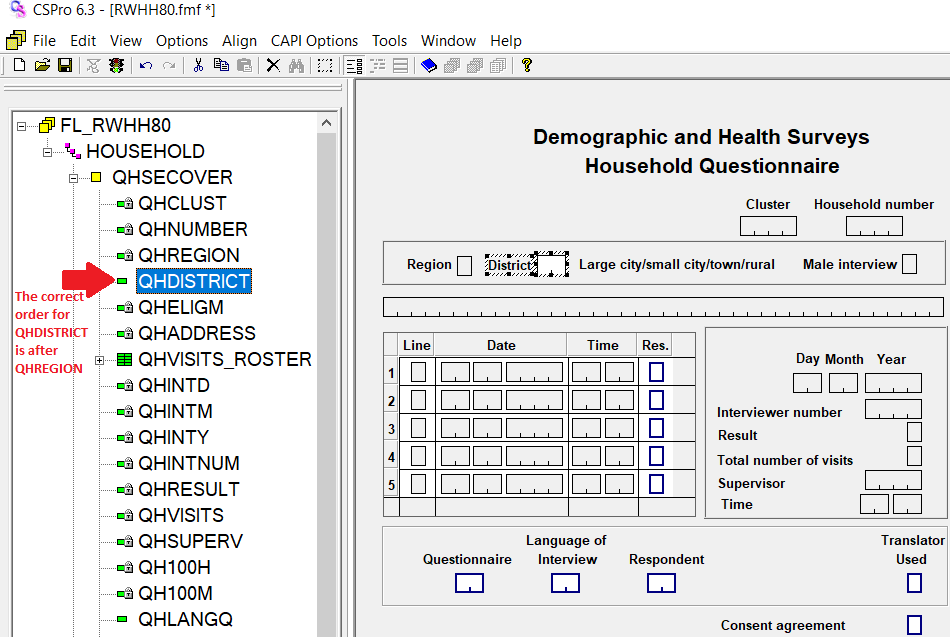
**Step 2:** Start modifying your forms from the very top form. Switch to the dictionary tab on the left panel and drag any country specific variables you added in the dictionary that will be placed on the first form. Country specific variables not dragged in the form appear in blue in the “Dicts” tab and those already dragged into the form appear in grey.



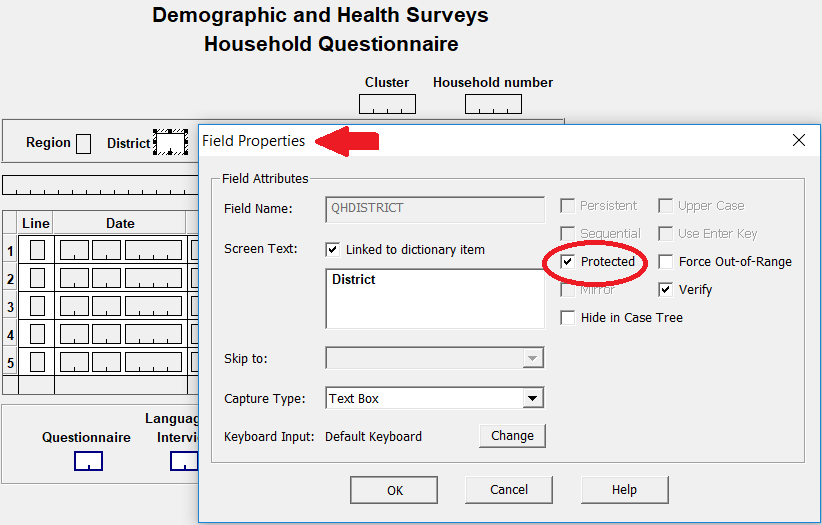
**Step 3:** After dragging your country specific variables to the form, they are placed at the end of that form in the tree (left pane).

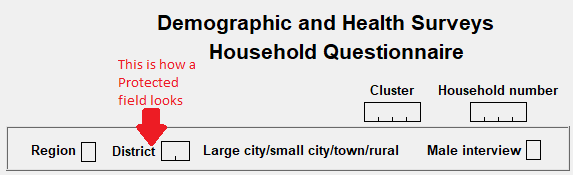


Switch to the “Forms” tab in the left panel and click and drag the country specific variable in the correct order it need to be.

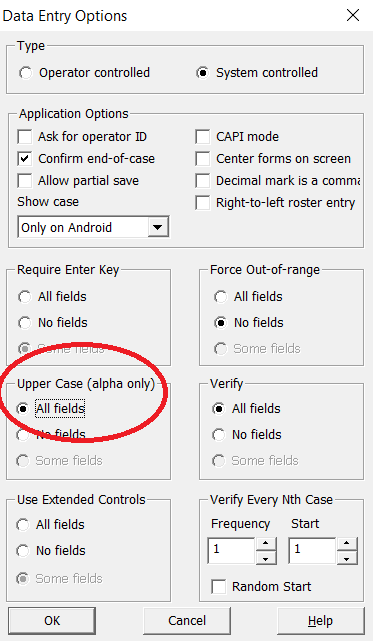


**Step 4:** Right click the country specific variable dragged into the form to set its “Field properties” to what is needed. Could be the type of extended controls used under “Capture type” or other property like “Protected” or “Upper Case” (this last mandatory if the country specific variable is Alpha). In our example, QHDISTRICT need to be a “Protected” field, and will be managed through the logic.

****

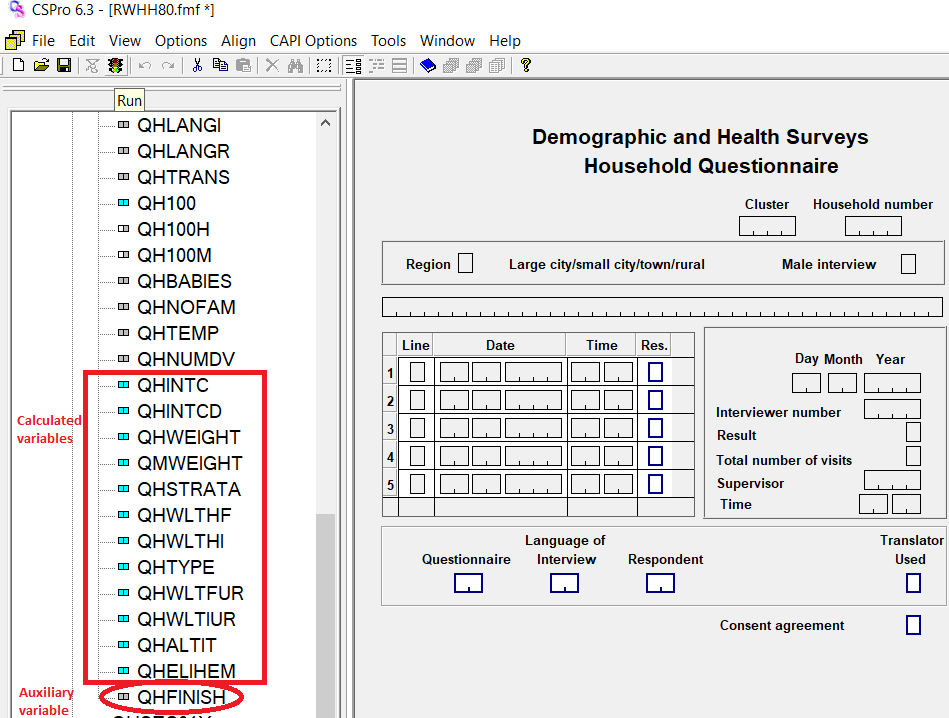
****

Can also set “Upper Case” for alpha variables from the global “Options” menu >> “Data Entry…”, and then tick “All fields” under “Upper Case” section, as shown below:

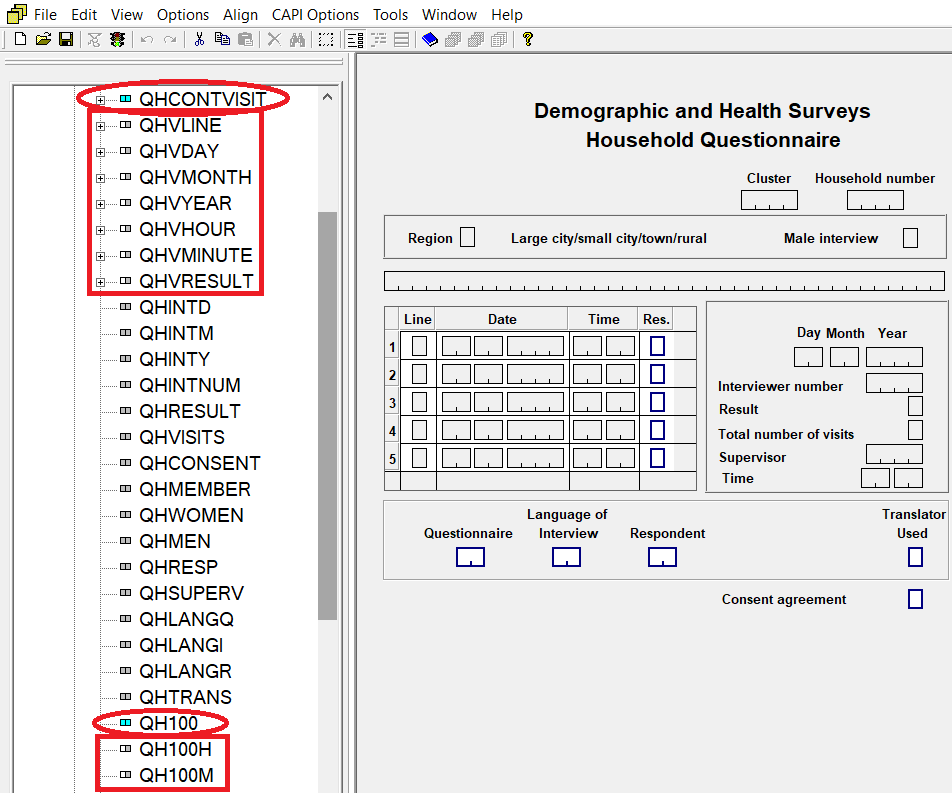
****

Please note that in order to set the “capture type” of CAPI extended controls for alpha variables as “check box”, the length of the alpha variable has to equal the total number of letters in the question. For example, in an alpha variable having the categories ABCDEFXZ, the total number of letters are 8, thus the length of that variable in the dictionary will be 8.

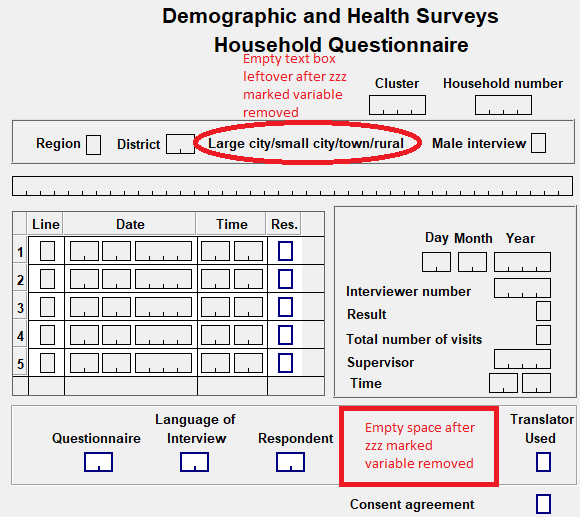
**Step 5:** The calculated variables described in the .IN file manual will appear in blue, but none of them need to be dragged into the forms as they will not be entered during field operations, instead they will be calculated at a later stage after the data have been cleaned. On the contrary, auxiliary variables that helps controlling the CAPI data collection programs will appear in grey, as they are used during CAPI data collection. On the image below, the left panel shows calculated and auxiliary variables present in the first record of the dictionary. Calculated variables appear in blue because they are not needed during data collection, but the auxiliary variable appears in grey.



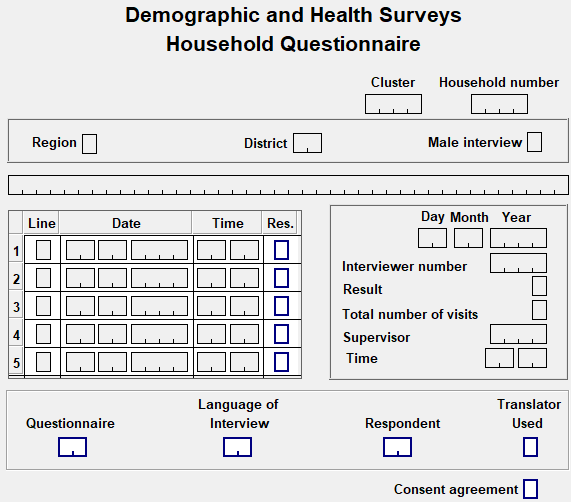
**Step 6:** Note that when subitems are present, they will be placed in the form instead of the items and the last will remain in blue in the dictionary record. This is Ok because we will be capturing information for each subitem during data collection, thus no need to place the items in the form.



**Step 7:** Inspect the form and look for empty spaces or leftover text boxes that remained after zzz marked variables were refreshed/removed from the form. You will need to redesign the form to fill up empty spaces with existing variables in the form. Also, properly align your variable to their text to facilitate a tidy form view for interviewers. Always mimic the questionnaires when re-aligning or re-designing.



This is how the first form looks after re-aligning and re-designing:



**Step 8:** Repeat steps above for all subsequent forms on the Household Form file. Then, continue working these steps in your CAPI Individual Form file C:\CC81\**Entry**\**CC**IN**8**0.fmf. Save your changes OFTEN!

# 4. Adjusting the Logic for the CAPI Data Entry Programs

## 4.1 What is New in CollectHH Application Logic

### 4.1.1 The PROC GLOBAL

The PROC GLOBAL is the first procedure in CSPro logic applications. There you will find declaration of temporary/working variables, arrays, and functions. In the DHS8 application, most working alpha variables have been declared as string, because string automatically resizes to match the alphanumeric value assigned, whereas variables declared as alpha has a fixed-width. Let us highlight some “what’s new” in this CSPro 7.4 and DHS8 questionnaire CollectHH.

*+new*

string strnotes;            { for editnote at the end of the interview }

strnotes allows to take or change a note at the end of the CollectHH interview when the interview is being added for the first time (adding mode). It is used in variable QHFINISH.

*+new*

array string DaysWeek(7);   { to store the days of the week }

string dayalpha;             { literal for day of the week }

The DaysWeek string array is used in conjunction with function initializeDays that provides weekdays in English. The user-defined initializeDays function uses the CSPro function “tr” that translates text in the language used by the application. Text translations are stored in CollectHH message’s file.

{ set value sets based on language }  
function InitializeDays()  
  DaysWeek(1) = tr("Monday");  
  DaysWeek(2) = tr("Tuesday");  
  DaysWeek(3) = tr("Wednesday");  
  DaysWeek(4) = tr("Thursday");  
  DaysWeek(5) = tr("Friday");  
  DaysWeek(6) = tr("Saturday");  
  DaysWeek(7) = tr("Sunday");  
end;

Below is an example of the message file with translated texts into Spanish that can be used by the “tr” function when setlanguage is declared as Spanish, i.e., setlanguage( “ES” ); :

{ common strings used by data collection applications }  
  ES ( "Not a household member" ) No es miembro del hogar  
  ES ( "father" ) padre  
  ES ( "mother" ) madre  
  ES ( "Fieldworker" ) Trabajador de campo  
  ES ( "Household" ) Hogar  
  ES ( "Births" ) Nacimientos  
  ES ( "Siblings" ) Hermanas(os)

DaysWeek and dayalpha are used in the Child Labor Module of the CollectHH application.

*+new*

array chfound(50);          { identify if a child in biomarkers is found in the woman's birth history }

array birthd(50);

array birthm(50);

array birthy(50);

alpha(2) xline;

We are now searching for the date of birth of the child from the mother’s individual interview if she is an eligible woman with an available interview recorded. Then we are using the mother reported date of birth, store it in the arrays birthd, birthm, and birthy, and assign them to the biomarker date of birth variables for that child. This is the reason we are now passing the individual data file as an external look up file in the household data capture program. The logic to accomplish this is below:

PROC QBSEC01  
preproc  
  { initialize array to store children's data birth coming from women's questionnaire

if available }  
  do i = 1 while i <= 50  
    chfound(i) = 0;  
    birthd(i)  = 0;  
    birthm(i)  = 0;  
    birthy(i)  = 0;  
  enddo;  
  { load date of birth for all children of eligible women in the household }  
  do i = 1 while i <= QHMEMBER  
    xline = edit("99", QH09(i));  
    if QH09(i) & loadcase( ccIN80, QHCLUST, QHNUMBER, xline ) then  
      do j = 1 while j <= soccurs(QWSEC2B)  
        if Q227(j) in 1:QHMEMBER then  
          chi = Q227(j);   
          chfound(chi) = 1;  
          birthd(chi)  = Q220D(j);  
          birthm(chi)  = Q220M(j);  
          birthy(chi)  = Q220Y(j);  
          if validyr(Q220Y(J)) & QBINTY - Q220Y(j) > 5 then birthy(chi) = 9998 endif;  
          exit;  
        endif;  
      enddo;  
    endif;  
  enddo;

Then, the logic for copying the day of birth, just as one example of the three, is this:

PROC QB103D  
preproc  
  set attributes ( $ ) native;  
  if chfound(QB102) then  
    set attributes( $ ) protect;  
    $ = birthd(QB102);  
  endif;

*+new*

numeric android = 0;        { switch that may be used once the application runs in

Android OS }

This switch is currently being used on the field “Any other person living in household”, in Android mode only, when displaying the list of the household members collected so far (name, relationship, sex, and age). As a reminder, in Android data capture the household members questions are shown one at a time thus it makes sense to display the list of all household members entered when asking “are there more members in the household”. In Windows, it is not necessary to show the list of all household members because one can see the roster of household members just on top of the screen. The *showhh* function below is new to the household data capture application, but it existed in the individual data capture application since CAPI Programs exist.

{ help function to display the household roster }  
function showhh()  
  show( QHSEC1X\_ROSTER, QHFIRSTN, QHRELAT, QHSEX, QHAGE  
      title( tr("Name"), tr("Relation"), tr("Sex"), tr("Age") ) );  
end;

*+new*

valueset individuals;

Individuals are dinamically created valuesets defined in the PROC GLOBAL using the statement valueset and defined in this case as a numeric valueset. It is used in conjunction with the setvalueset function to dynamically change the specific item valueset and the item type must match the valueset type. One can declare valuesets as string by explicitly indicating “string” in the valueset declaration.

Individuals valueset is used all along the CollectHH and CollectIN data capture applications whenever a member line number/name is needed, for example the variable QHRESP – Line number of household respondent:

PROC QHRESP  
onfocus  
  individuals.clear();  
  do i = 1 while i <= QHMEMBER  
    if QHAGE(i) in 12:98 then  
      individuals.add(QHFIRSTN(i) + " " + QHLASTN(i), i);  
    endif;  
  enddo;  
  SetValueSet( $, individuals );

Other variables where valueset individuals is used are:

QH13 – Mother’s line number

QH15 – Father’s line number

QH105 - Line number of person who collects water

QH146A to QH146D – Line number of n person who slept under net

QB118 – Line number of parent/caretaker

QB127 – Line number of parent/caretaker

QB317 – Line number of parent/caretaker

### 4.1.2 GPS Collection in the Household Data Capture Application

*+new*

GPS collection in the household data capture application

Now, right before the end of the household interview, the interviewer has the option to replace the household coordinates already collected during the listing operation. He is prompted to two questions:

Coordinates for the household were already collected, do you want to replace the current coordinates?

And a second confirmatory question if the response to prior question was ‘Yes’:

Are you sure that you want to replace the current coordinates?

If the interviewer decides to re-collect GPS coordinates, s/he will insert the GPS dongle and the program will automatically collect latitude, longitude, altitude, and accuracy (1-the best, 50-the weakest).

### 4.1.3 Warning Instead of Errmsg

*+new*

Warning instead of Errmsg

The Warning function only has special behavior in the data entry application; in batch applications it works identically to Errmsg. During data entry, Warning messages are not displayed to interviewers when advancing in a case or resuming from a partially saved case. A good number of data entry messages have been changed from Errmsg to Warning. Some Warning messages are accompanied by the ‘select’ option, that prompts the interviewer to either correct one piece of the data (age or relationship) or continue. One example of a Warning message in CollectHH is as follows:

PROC QHAGE  
  { minimum age for head and spouse }  
  if QHRELAT in 1,2 & QHAGE < MinAgeHS then  
    warning( 0071, $, strip(getlabel(QHRELAT, QHRELAT)), MinAgeHS )  
      select( tr("Relationship"), QHRELAT, tr("Age"), $, tr("Continue"), continue );  
  endif;

Where: MinAgeHS = 12; { Min Age to be head/spouse – See dictionary level preproc }

In Message file (.Mgf):

0071 W Age (%02d) of %s is under %02d

### 4.1.4 Function tr

*+new*

Function tr

This new CSPro function is widely used in CollectHH to translate logic string literals in the data capture application (default is English) into other languages (Spanish, French…). Translation of string literals are stored in the data capture message file (.Mgf). See example below:

PROC QHSEX  
  if curocc() > 1 & QHRELAT = 2 then  
    if $ = $(1) then  
      errmsg( 0060, $, $(1) )  
        select( tr("Sex of head"), $(1), tr("Sex spouse"), $ );  
    endif;  
  endif;

Then, in message file (.Mgf):

{ common strings used by data collection applications }

:

ES ( "Sex of head" ) Sexo jefe  
ES ( "Sex spouse" ) Sexo conyugue

### 4.1.5 Messages in Multiple Languages (in .MGF file)

*+new*

Messages in multiple languages (in .Mgf file)

In CSPro 7.4 we can define multiple messages for a given message number using the language directive equals language name, followed by the message number and the message. CSPro will automatically display the correct message based on the current language. If no translated message exists for the current language, the default message is shown. Using the language directive declares that all subsequent messages are in a given language, for example:

language = EN  
 0001 E Cluster %04d not found in the sample design file  
 0002 E Filename for %s file incorrect (%s): Quit, correct the filename and restart

language = ES  
 0001 E Conglomerado %04d no encontrado en el archivo de diseño de la muestra  
 0002 E Nombre de archivo %s incorrecto (%s): Cierre, corrija el nombre de archivo

y reinicie

### 4.1.6 Blocks

*+new*

Blocks

Blocks are a way to group several fields into a related unit. Fields in a block should come from the same record, group, or roster. Blocks can have question text, which is ideal for multipart questions like quantity/unit, day/month/year, household assets, for example, “Does your household has”, etc. Logic can be defined in the block procedure for all related fields within the block, instead of having the logic in the field procedures. You can refer to block names in the logic for movement statements ([advance](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/advance_statement.html), [ask](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/ask_statement.html), [move](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/move_statement.html), [reenter](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/reenter_statement.html), [skip](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/skip_statement.html)) and in occurrence-related functions ([count](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/count_function.html), [curocc](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/curocc_function.html), [maxocc](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/maxocc_function.html), [noccurs](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/noccurs_function.html), and [totocc](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/totocc_function.html)). A block can exist without any fields added to it, in this case the block acts as a “control field” and its logic is executed in the order where the block is placed in the Form Tree. This is ideal for filtering questions and/or orienting your skips within the application.

In the CollectHH application, questions in blocks are:

* Household chores by children in the Child Labor record
* Ownership of animals, Household possessions, and Possessions of household members in the Household Characteristics record
* Child’s date of birth, Date of measurement in the Weight, Height, and Hemoglobin Measurement for Children Age 0-5 record
* Date of measurement – woman in the Weight, Height, and Hemoglobin Measurement for Women record
* Date of measurement in the Weight, Height, and Hemoglobin Measurement for Men 15-49 record

### 4.1.7 The DHS Program Developed Functions That Are Now Part of CSPro

*+new*

Functions AdjLBA, AdjUBA, AdjLBI, AdjUBI, SetLB, and SetUB are now part of CSPro

The DHS Program developed functions to adjust or return the lower/upper bound of the century months code (CMC) of an event are now part of CSPro, thus no need to declare them in the DPCollect.apc file in the \Library folder. Let us see the description of use of these functions.

AdjLBA/AdjUBA: adjusts the lower/upper bound of the *century month code* (CMC) of an event based on an age. It is generally used to calculate the minimum/maximum date of birth based on the date of an interview.

AdjLBI/AdjUBI: adjusts the lower/upper bound of the *century month code* (CMC) of an event based on an interval. It is generally used to calculate the minimum/maximum *next*/*previous* date of birth based on a date of birth and a minimum number of months allowed between births.

SetLB/SetUB: returns the lower/upper bound of the *century month code* (CMC) of an event.

### 4.1.8 Recode/Endrecode Statement Instead of Box/Endbox Statement

*+new*

Recode/Endrecode statement instead of Box/Endbox statement

Starting with CSPro version 7.4, there is a newly define recode/endrecode statement that is more powerful than the prior recode or box statements, in the sense that this one can assigns a value to one or more output variables based on the value of one or more other input variables. Input variables are evaluated to either a number or string. When using multiple input or output variables, they should be separated by two colons (::), not one as the previous recode or box statements. The output variables can be dictionary items, working variables, array values, list values, or the return values of a user-defined function. An output variable can also be included among the input variables. Let us see few examples of the recode statement below:

Example where the output variable is a working variable and the input variable AGE is evaluating the resulting values of an arithmetic operation (subtraction):

PROC AGE  
  
    numeric valid\_age\_difference\_with\_head;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Recode | RELATIONSHIP | :: | AGE(1) – AGE | -> | valid\_age\_difference\_with\_head; |
|  | 3, 4 | :: | <  12 | -> | false; // biological and step children must be 12+ years younger |
|  | 6 | :: | > -12 | -> | false; // parents must be 12+ years older |
|  | 7 | :: | <  24 | -> | false; // grandchildren must be 24+ years younger |
|  |  |  |  | -> | true; |
| [endrecode](mk:@MSITStore:C:\Program%20Files%20(x86)\CSPro%207.4\CSPro.CHM::/recode_statement.html); |  |  |  |  |  |

Example where the output variable is the return values of a user-defined function:

function string GetDisplayName()

|  |  |  |  |
| --- | --- | --- | --- |
| recode | NAME | -> | GetDisplayName; |
|  | "" | -> | "(Undefined)"; |
|  |  | -> | strip(toupper(NAME)); |
| endrecode; |  |  |  |

end;

Example of a string input variable and multiple resulting output variables:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Recode | INDUSTRY\_SECTION | -> | min\_division | :: | max\_division; |
|  | "A" | -> | 1 | :: | 3; |
|  | "B" | -> | 5 | :: | 9; |
|  | "C" | -> | 10 | :: | 33; |
|  | // ... |  |  |  |  |
| endrecode; |  |  |  |  |  |

In the CollectHH application, the recode statement is used in the Biomarker questionnaire to:

* check the weight in kilograms by sex and age in months for children 0-72 months old (PROC QB106)
* check the height in centimeters by sex and age in months for children 0-72 months old (PROC QB108)

### 4.1.9 New variables added as part of DHS-8 Questionnaire Revision

*+new*

New variables added as part of DHS-8 Questionnaire Revision

#### 4.1.9.1 Household Questionnaire

Reliability of water source revised:

* Wording change in 106, DHS-7: In the past two weeks, was the water from this source not available for at least one full day?; DHS8: In the last month, has there been any time when your household did not have sufficient quantities of drinking water when needed?

Emptying of septic tanks and latrines:

* Filter 113, if type of toilet facility are 12 (Flush to septic tank), 13 (Flush to pit latrine), 21 (Ventilated/improved pit latrine), 22 (Pit latrine with slab), 23 (Pit latrine without slab/open pit) or 31 (Composting toilet) then ask questions 114-116, otherwise skip to question 117
* 114, Has your (septic tank/pit latrine/composting toilet) ever been emptied?
* 115, The last time the (septic tank/pit latrine/composting toilet) was emptied, was it emptied by a service provider?
* 116, Where were the contents emptied to?

Proportion of population with reliance on clean fuels and technologies

Cook stoves:

* 117, In your household, what type of cookstove is mainly used for cooking?
* 118, Does the stove have a chimney?
* 119, Does the stove have a fan?
* 120, What type of fuel or energy source is used in this cookstove?

Fuel source for space heating:

* 123, What does this household use to heat the home when needed?
* 124, Does it have a chimney?
* 125, What type of fuel or energy source is used in this heater?

Fuel source for lighting:

* 126, At night, what does your household mainly use to light the home?

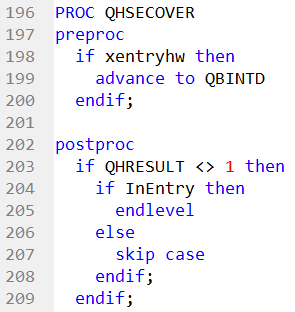
Mobile money:

* 135, Does any member of this household use a mobile phone to make financial transactions such as sending or receiving money, paying bills, purchasing goods or services, or receiving wages?

## 4.2 Good Programming Practices

### 4.2.1 Indenting the Code

Indenting the code make it easy to read a program. It is a good idea to always indent the code. The code in all standard programs in the DHS surveys are indented, as shown in the example below. PROC, preproc, and postproc statements usually start in the very first column. Statements that follows are indented with two spaces, followed by additional two indented spaces for nested statement that follows.



### 4.2.2 Documenting the Program by Adding Comments

Comments make applications easier to understand. They are used to explain the purpose of specific statements or to temporarily disable statements to help find errors. Any text enclosed by braces { } is a comment. The text within the brackets will be green. Comments are not checked for syntax errors. Any text following double slashes // until the end of the line is also a comment.

### 4.2.3 Other Good Practices

* Write CSPro statements in lowercase.
* Write variable names from the dictionary in UPPERCASE.
* Use functions when an operation is generalizable or if the same operation will be called repeatedly from different procedures.
* Use standard prefixes for working variable names. For example: cnt: counts of an item (such as cntMembers, cntHouseholds, etc.)

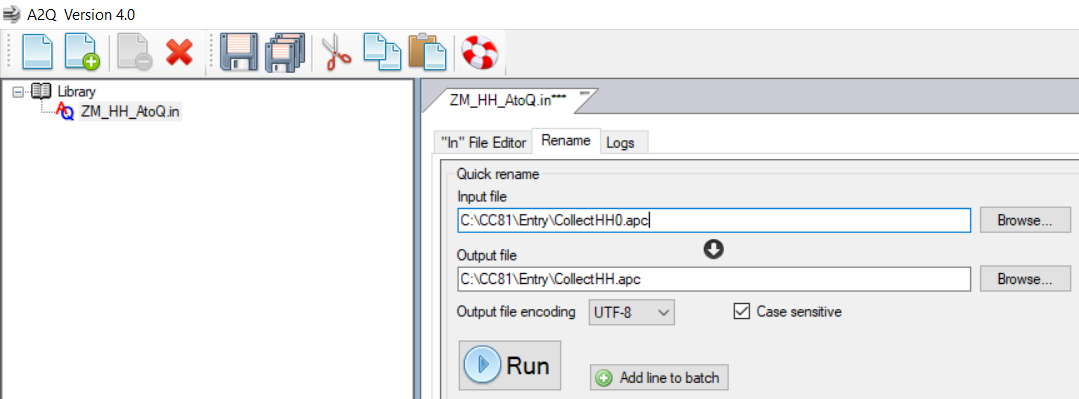
## 4.3. Adjusting the Logic for the Data Entry Programs CollectHH & CollectIN

We are using the DHS8 standard applications to present the content of this documentation and we will explain logic examples based on the DHS8. EXPLAIN OR NOT WE ARE USING DHS7 DICTIONARY for the A2Q because at the time of writing this manual there was no DHS8 survey in the programming phase.

Adjusting the data entry programs’ logic consist of converting the standard applications into country specific application. The CAPI data capture consists of two applications, one for the household –CollectHH, and one for the individual questionnaire –CollectIN. The first step to convert the applications is to use the A2Q program to convert standard variables named Axxx into Qxxx in the .apc files. For the purpose of the logic and going forward, we are going to use the Zambia DHS 2018 survey files for our examples. The reason being it is a DHS, a more complete survey, rather than the Ghana MIS files we used with the IN file, dictionary, and form documentation. Let us see the conversion of the standard applications using the IN files from Zambia DHS 2018 and the A2Q program.

### 4.3.1 Run A2Q on Application Files

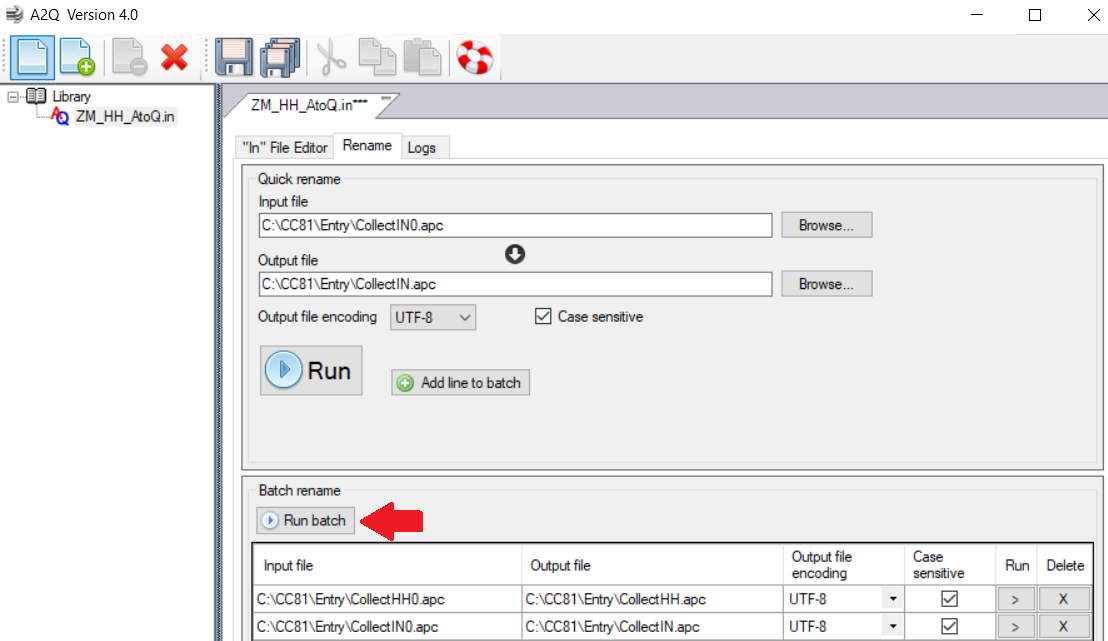
Open the Household IN file for Zambia DHS 2018 and click on the “Rename” tab. As the input file, use \CC81\Entry\CollectHH0.apc, which I previously renamed from CollectHH.apc into CollectHH0.apc, so that I can use filename CollectHH.apc as output file. Then, as output file, use \CC81\Entry\CollectHH.apc. Use UTF-8 as “Output file encoding”, keep the box checked contiguous to “Case sensitive”, and then click on “Add line to batch”.



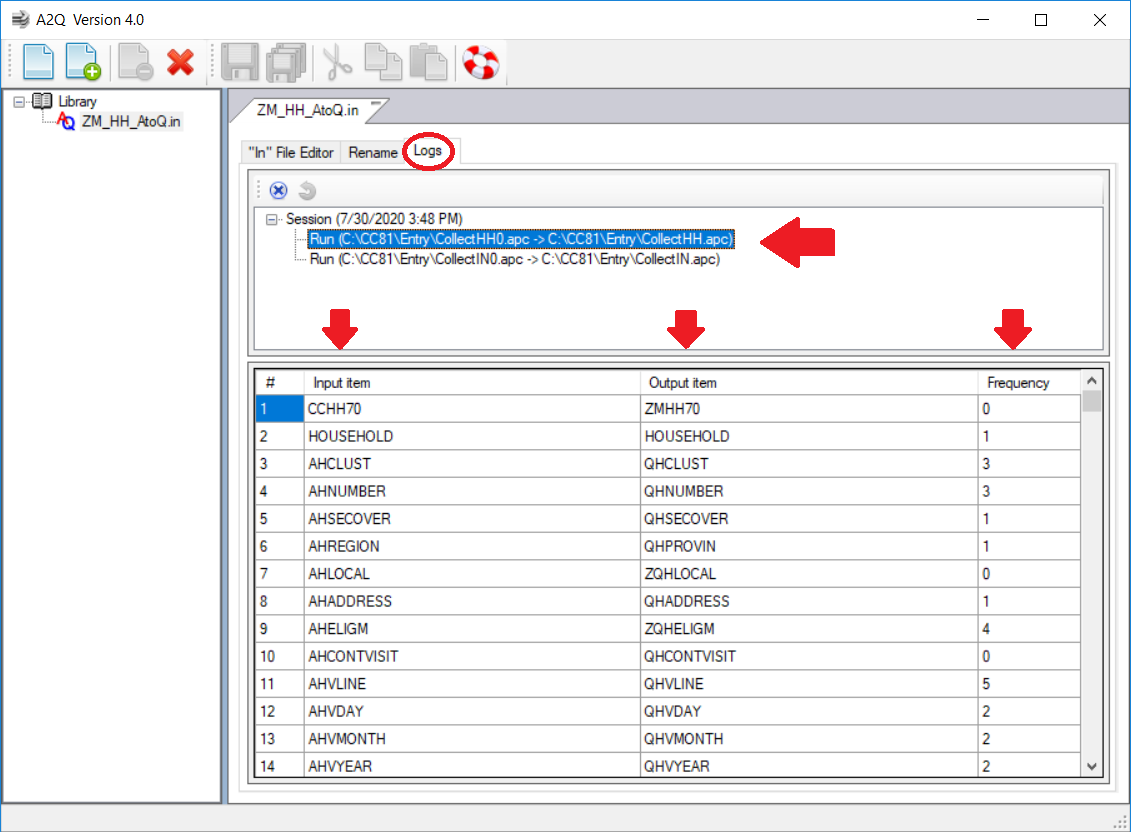
Similarly, we will do the same steps but this time to add the Individual application file. As the input file, use \CC81\Entry\CollectIN0.apc, which I previously renamed from CollectIN.apc into CollectIN0.apc, so that I can use filename CollectIN.apc as output file. Then, as output file, use \CC81\Entry\CollectIN.apc. use UTF-8 as “Output file encoding”, keep the box checked contiguous to “Case sensitive”, and then click on “Add line to batch”.

Then, click the “Run as batch” button.

Note: CC are the 2-letters for the country code or abbreviation.



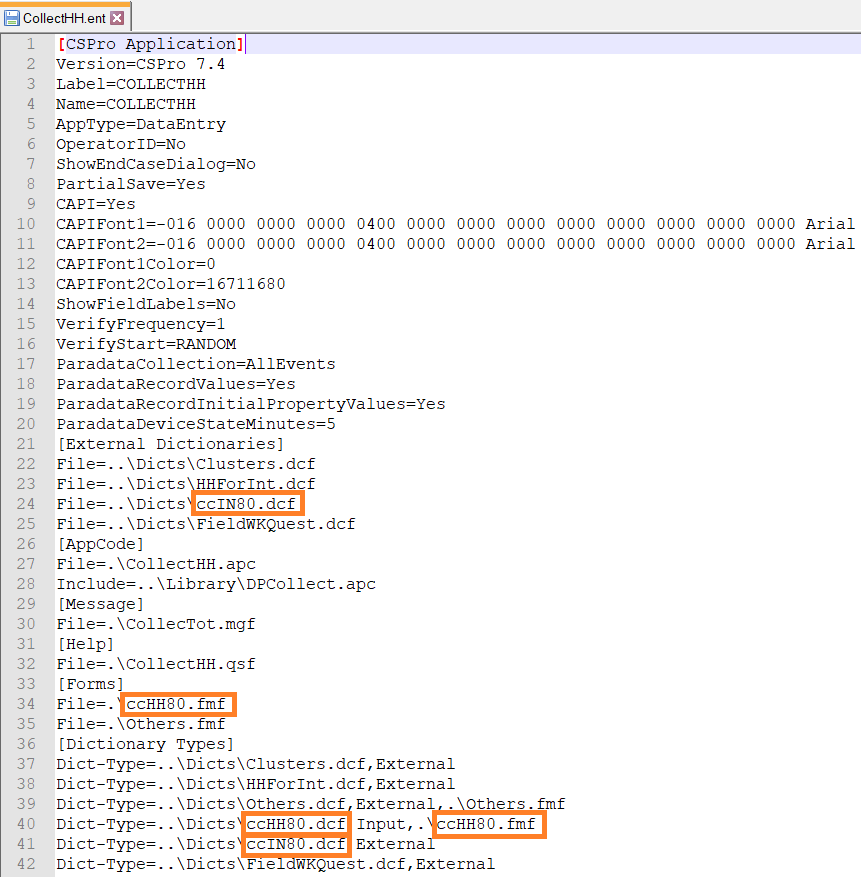
Next, the A2Q program shows in the “Logs” tab the frequency of each variable that got replaced its name from A to Q or from A to Z when the variable is not used in the survey questionnaire. The log for each batch run (CollectHH and CollectIN) is available at the upper center part of the window under “session”; and to toggle between the two logs, just click on them.



### 4.3.2 Household Application – COLLECTHH

#### 4.3.2.1 Modifying Filenames on .Ent File to Open CSPro Household Application

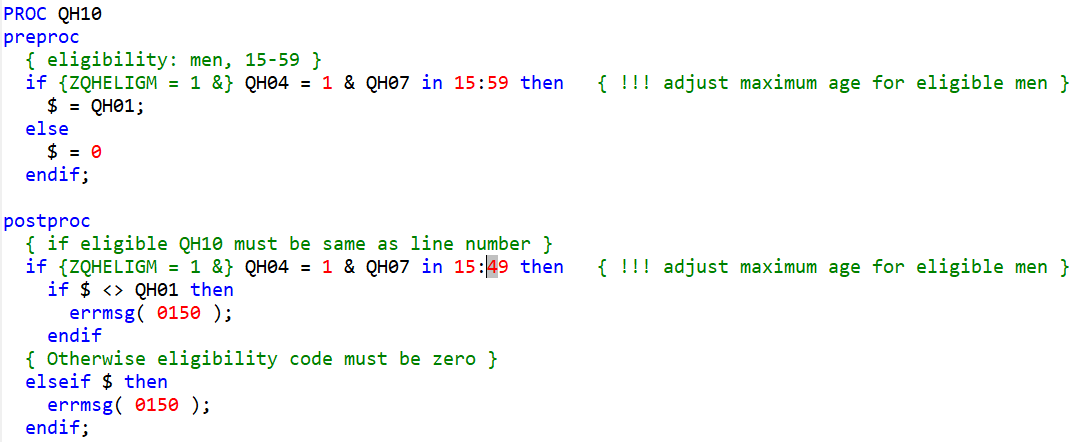
In CollectHH.ent you will modify filenames to reflect the names from your country files. CC are the 2-letters for the country code or abbreviation. Other files like “External dictionaries” Clusters, HHForInt, FieldWKQuest remain with the same filename and will be discussed in a separate session. CollectHH will also need external dictionary ccIN80, which is the Individual questionnaire dictionary loaded as an external file, the external logic file DPCollect stored in the \Library folder, the message file CollectToT which is the same message file used for both CollectHH and CollectIN, and the .Qsf file that we will discuss in another session. Form files used in this application are the ccHH80.fmf containing the main screens for the survey questionnaire and the Others.fmf to enter the other responses.



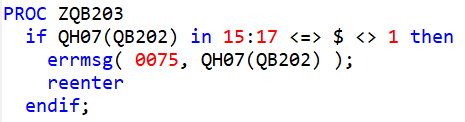
#### 4.3.2.2 Opening CollectHH.ent

At this point you can open the household application CollectHH.ent. Do not compile the application. The first thing you will do is scan your logic application from top to bottom identifying ZZZ or Z variables that are not applicable. If this is the first time you are working with CollectHH logic application, you will comment out pieces of logic where the ZZZ or Z variables are found or you will comment out the entire procedure (see examples below). If you are a master on modifying DHS CAPI data capture applications, you can delete the pieces of logic where ZZZ or Z variables appears if you are comfortable doing that. Other instruments you will need are a printed copy of your survey questionnaires and the DHS standard questionnaires electronic files or a printed copy, to check filters, skips, eligibility criteria for a module in your survey questionnaires against the DHS standard questionnaires.

Example of a commented-out logic within a procedure: in many DHS surveys, man’s interviews are conducted on a subsample of households. But, in this example survey, man’s interviews were administered in all households, thus variable ZQHELIGM that controls the household subsample for man’s interview is not applicable and its logic is commented-out. Another adjustment needed in procedure QH10 is setting the maximum age for eligible men (as read on the comment), in this survey eligible men are those between 15 and 59 years old. The adjustment is needed because the standard logic’s default is for men 15 to 49 years old.



Example of a commented-out procedure: the procedure ZQB203 does not exist in this example survey even though variables in the logic within the procedure do exist and the logic can be evaluated. Comment out or delete the entire procedure.



Once you have scanned your application from top to bottom and taken care of the ZZZ or Z variables, then you may compile your application. There may still be few compilation errors that you will need to take care.

#### 4.3.2.3 Customary CAPI Program Functions

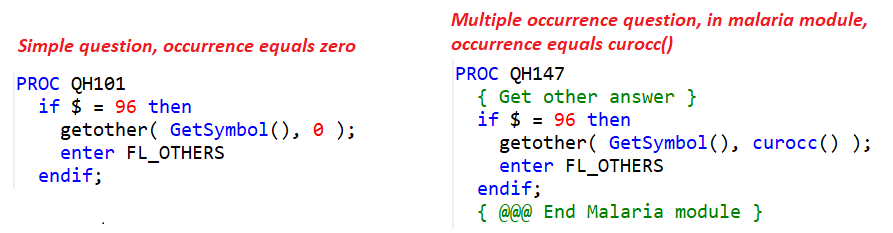
In this section we are going to list other functions in the PROC GLOBAL that have existed since our CAPI Programs exist or even before since paper questionnaire data processing exist. You should get familiar with them and they are available in the CollectHH application:

function userbase(); { setup basic user bar }

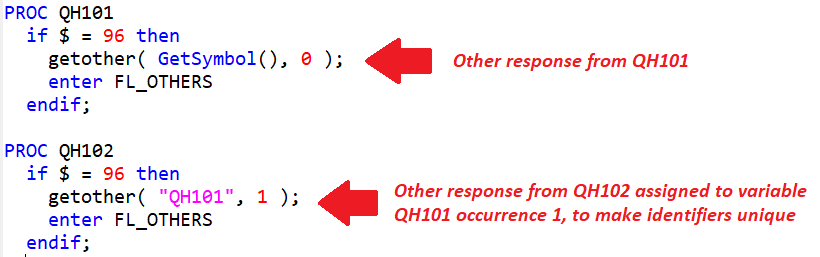
function OnChangeLanguage() { set value sets based on language }

function getother( alpha(10) questnum, questocc ); { Function to ensure proper entry and updating of "Other" answers }

Please note that this function has two parameters that need to be passed when used: the question name or number and the occurrence. Every time we call this function in a multiple occurrence record, please remember to indicate the occurrence in the second parameter. Many times, this is forgotten making impossible to retrieve ‘other’ responses for a specific question in the future. See below differences of using getother in a simple question versus a multiple occurrence question.



On few occasions, the ‘other’ response of a variable is stored as an extension of the ‘other’ response from a prior variable. For example, the ‘other’ response given in QH102, is stored as occurrence #1 (an additional occurrence to make identifiers unique) to variable QH101:



function updthhassgn() { update file with households assigned to interviewers }

#### 4.3.2.4 Looking for Code to Modify (!!!) and Adjusting Parameters of Temporary Variables

Throughout the data capture applications, there are comments starting with exclamation marks (!! or !!!). These marks indicate that you will need to update the code encircled within it according with your survey details. Updates will be needed in specific code or in temporary variables parameters. For example, in the CollectHH application you have the following exclamation marks for which updates are needed in the logic:

* Under preproc of PROC FL\_ccHH80:

if cmcode(mm,yy) < cmcode(1,2019) then    { !!! Fix minimum date of interview }  
    errmsg( 0003, dd, mm, yy );  
    endlevel;  
  endif;

Some parameters may not be needed and can be commented out or deleted. For example, *survyear* parameter below is only applicable for the child labor module; if your survey does not have the child labor module, you can comment *survyear* out or delete it.

healthy  = 2014;  { !!! Year of start of health section }

survyear = 2020; { !!! change to year when the survey was launched (for child labor module) }

firstday = 3; { !!! change to day for the first day on survey year (1-Monday, 2-Tuesday,..7-Sunday) }

* Other PROCs:

PROC QHREGION  
preproc  
  { !!! Set region coming from sample file }  
  if demode() = add then  
    $ = YREGION;  
  endif;  
  
PROC QHTYPE  
preproc  
  { !!! Set locality or urban/rural coming from sample file  }  
  {     check field tables require an Urban/Rural definition }  
  if demode() = add then  
    $ = YURBRUR  
  endif;

PROC QHTEAM  
preproc  
  { !!! make sure to properly assign team number based on supervisor code }  
  { code here assumes that first two digits are the team number and that  }  
  { codes start with 1000 for first team                                  }  
  $ = int(QHSUPERV/100) - 10 + 1;

PROC QHMEN  
preproc  
  if QHELIGM = 1 then

{ !!! adjust maximum age for eligible men }  
    $ = count( QHSEC1X\_ROSTER where QHAGE in 15:59 & QHSEX = 1 );

  else  
    $ = 0;  
  endif;

PROC QH10  
preproc  
  { eligibility: men, 15-59 }

{ !!! adjust maximum age for eligible men }  
  if QHELIGM = 1 & QH04 = 1 & QH07 in 15:49 then

    $ = AH01;  
  else  
    $ = 0  
  endif;  
  
postproc  
  { if eligible AH10 must be same as line number }

`{ !!! adjust maximum age for eligible men }  
  if AHELIGM = 1 & AH04 = 1 & AH07 in 15:49 then     
    if $ <> AH01 then  
      errmsg( 0150 );  
    endif

PROC QH131  
{ !!! make sure to add code to properly check the units for different types of land }  
{     with the corresponding area.  Also make sure that if units are used, code 95  }  
{     should be accompanied to the unit it refers to. i.e., 9-95 isn't useful       }

Note: see also section “**Adding Logic to Check Land Area”** in this manual.

PROC GHINTRO  
postproc  
  { open the GPS dongle to capture GPS coordinates }  
  { !!! verify using the device manager the port number used by the dongle       }  
  {     in ICF laptops port 3 is reserved and erroneously taken as a dongle port }  
  {     in ASUS tablets ports are assigned starting from port 3 and above        }  
  capturecoor = ( visualvalue(GHLATITUDE) = notappl );  
  if $ = 1 & GHAUX <> 2 then  
    do i = 3 while i <= 10  
      if GPS( open, i, 4800 ) then  
        break  
      endif;  
    enddo;

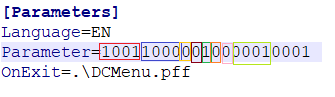
Note: if using ICF laptop, start the loop in i = 4 to avoid Windows taking the reserved port 3 where the GPS dongle do not work.

#### 4.3.2.5 Parameters Coming from DCmenu Stored at CollectHH.pff File

The parameters below are stored in the CollectHH.pff at the time of execution of the data entry menu and are dynamically loaded into the data capture application to take actions. Check that you properly call them in, for example, if your survey does not have Height and Weight, would you need to modify the application to reuse position 13 for another value?

{ converts parameter to numbers  }  
xintnum  = tonumber( sysparm()[1:4] );    { interviewer number }  
xsupnum  = tonumber( sysparm()[5:4] );    { supervisor number }  
xmodcent = tonumber( sysparm()[9:1] );    { modify by central office }  
xrvisit = tonumber( sysparm()[10:1] );   { revisit HH }  
xmodify   = tonumber( sysparm()[11:1] );   { modify completed HH }  
xpartial = tonumber( sysparm()[12:1] );   { if case has been saved partially }  
xentryhw = tonumber( sysparm()[13:1] );   { enter height & weight data }  
xhhnum   = tonumber( sysparm()[14:4] );   { household number }

These parameters populate several fields in the data capture application, among other uses. The fields populated are: QHINTNUM from xintnum and QHSUPERV from xsupnum. A string with the parameter values is stored in the CollectHH.pff file:



Note: the last four digits in the Parameters string are the cluster number.

#### 4.3.2.6 Controlling Household Assignments per Interviewers

For every cluster, each interviewer in the team get assigned a number of households for data collection. The assignment is done by the Supervisor using the Supervisor’s menu and an assignment data file is transferred by the supervisor to each of the interviewers’ tablets. The data entry application uses this file to control the specific households for which the interviewer needs to collect data. The logic to make this control is under the dictionary level Preproc procedure. The first logic is to load the households assigned to the interviewer from the external household assignment file; if the household number passed from parameter *xhhnum* is not found, then an error message is displayed telling so.

PROC FL\_ccHH80

preproc

  { load households assigned to interviewers file }  
  if !xmodcent then  
    VNUMBER = xhhnum;  
    open( HHFORINT );  
    if !loadcase( HHFORINT, VNUMBER ) then  
      errmsg( 80004, VNUMBER );  
      endlevel  
    endif;  
  endif;

Once HHFORINT external file is open, its fields are used to populate several variables in the survey, such as the household eligibility for male interview (see PROC QHCLUST), the household number (see PROC QHNUMBER), the household address (see PROC QHADDRESS), and the household head’s name (see PROC AHFIRSTN).

At the end of a household interview, the HHFORINT data file is updated using function ‘updthhassgn’

when there is a change in the result of the household interview, or the interviewer code, or either the household head’s name or address. The ‘updthhassgn’ function is called from the postproc of the dictionary level FL\_ccHH80 as shown below.

PROC FL\_ccHH80

postproc  
  { update file with households assigned to interviewers }  
  if !xmodcent then    { central office doesn't update the sample assignation file }  
    updthhassgn();  
    stop(1);  
  endif;

#### 4.3.2.7 Sample Cluster File in the Data Entry Application

Similar to the households for interview file (HHFORINT), the cluster file is called from the Preproc of PROC FL\_ccHH80 to populate several fields. The function ‘loadcase’ is used to load the cluster file; if the cluster number entered in the Data Capture Menu system (obtained from the Parameter string in the CollectHH.pff file) is not found in the cluster file, then an error message is displayed telling so.

PROC FL\_ccHH80

preproc

if !xmodify then  
    YCLUSTER = tonumber( sysparm()[18:4] );  
    if !loadcase( CLUSTERS, YCLUSTER ) then  
      errmsg( 0001, YCLUSTER );  
      endlevel  
    endif;  
  endif;

The fields that are populated using the cluster file are: the cluster number (see PROC QHCLUST), region (see PROC QHREGION), and type of place of residence (see PROC QHTYPE).

#### 4.3.2.8 Countries with Polygyny

Polygyny, the most common form of polygamy, entails the marriage of a man with several women. In countries where polygyny is practiced, the household questionnaire will include an additional relationship to the head: co-wives referred in the standard logic with the code 9. Logic on polygyny must be uncommented under the procedure QHSEX.

PROC QHSEX  
  {{POLYG}  
  if QHRELAT = 9 & $(1) = 1 then   { !! Co-wives only for countries with polygyny }  
    errmsg( 0032 );  
    reenter  
  elseif QHRELAT = 9 & $ <> 2 then  
    errmsg( 0062 );  
    reenter  
  elseif QHRELAT = 9 & !count(QHSEC1X\_ROSTER where QHRELAT = 2 ) then  
    errmsg( 0063 );  
    reenter  
  endif;  
  {POLYG}}

In the message file:

0032 E For co-wives, head of household must be female  
0062 E Sex of co-wife not valid  
0063 E The presence of other wives requires that there should be one wife defined

before

Sometimes, the country practice polygyny, but Survey Managers did not include the code 9 = co-wives in the relationship question of the Household Schedule. Rather, code 2 = spouse will be used multiple times to equal all wives of the household head. If this is the case, you will need to adjust the following logic under QHSEX procedure:

  { Only one spouse, parent, parent-in-law of each sex expected }  
  if( count(QHSEC1X\_ROSTER where QHRELAT = 2 ) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 6 & QHSEX = 1) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 6 & QHSEX = 2) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 7 & QHSEX = 1) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 7 & QHSEX = 2) >= 2 ) then  
    errmsg( 0030 );  
    reenter  
  endif;

To

{ Multiple spouses, only one parent, multiple parents-in-law of each sex expected }  
  if( count(QHSEC1X\_ROSTER where QHRELAT = 2 ) >= 5 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 6 & QHSEX = 1) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 6 & QHSEX = 2) >= 2 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 7 & QHSEX = 1) >= 5 |  
      count(QHSEC1X\_ROSTER where QHRELAT = 7 & QHSEX = 2) >= 5 ) then  
    errmsg( 0030 );  
    reenter  
  endif;

#### 4.3.2.9 Passing Already Collected Information into the Permanent Household Roster

Information collected in the short household schedule QHSEC01X where household member names are asked are passed into the permanent household schedule using logic placed in the preproc of the PROC QHSEC01. Sometimes new country specific variables are added to the short QHSEC01X household schedule. In this case, do not forget to transfer the new country specific variables from the short QHSEC01X household schedule into the permanent QHSEC01 household schedule, by adding the logic in the preproc of PROC QHSEC01.

PROC QHSEC01  
preproc  
  { Initialize household members' questions with information already collected or

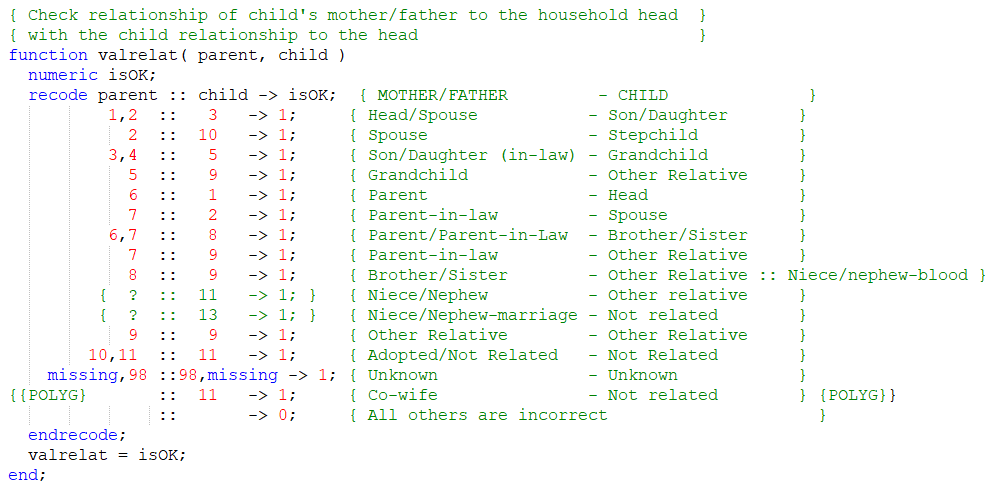
known }  
  do i = 1 while i <= maxmemb  
    if i <= QHMEMBER then  
      QH01(i) = i;

{ person's name must be added to the household schedule roster }  
      QH02(i) = strip(QHFIRSTN(i)) + " " + strip(QHLASTN(i));

      QH03(i) = QHRELAT(i);  
      QH04(i) = QHSEX(i);  
      QH05(i) = QHLIVES(i);  
      QH06(i) = QHSLEPT(i);  
      QH07(i) = QHAGE(i);  
    else  
      QH01(i) = notappl;      { blank out the entries not needed }  
    endif;  
  enddo;

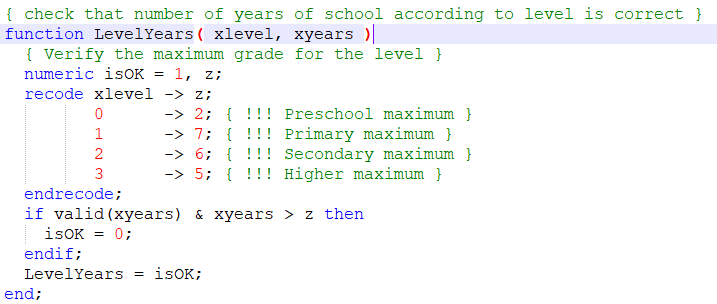
#### 4.3.2.10 Updating Function ValRelat with your Survey Specific Relationship Values

The ValRelat function is used in the Mother’s line number (QH13) and Father’s line number (QH15) variables of the Household Schedule. It resides in the DPCollect.apc file located in the \Library folder. This function checks valid relationship between the child’s parent to the household head with the child relationship to the head, otherwise it displays error message 0122. Sometimes, especially when there are survey specific relationship categories, you may need to add those categories to the function, otherwise you may get the error message 0122 triggered when the relationship is correct, in which case the secondary editor may attempt to correct the error and damage the information correctly collected in the field. Also, you will need to update function ValRelat to uncomment the logic when polygyny is practiced in the country and/or when there is a niece/nephew relationship category in your survey questionnaire.



#### 4.3.2.11 Updating Function LevelYears with your Survey Specific Relationship Values

The function LevelYears is used in the “grade of education” variables (in the household schedule these are QH17B and QH19B). It is stored in the \Library folder in the DPCollect.apc file. This function checks that the number of years of school according to level is correct. The values in z in this function are just a guide, you should update the values in z with the maximum number of years per level according to the survey-country educational system and add other levels and their maximum number of years per level according to the survey. For example, in a given country you may have Middle school in between Primary and Secondary levels, thus Middle school and the maximum number of years should be added to this function.



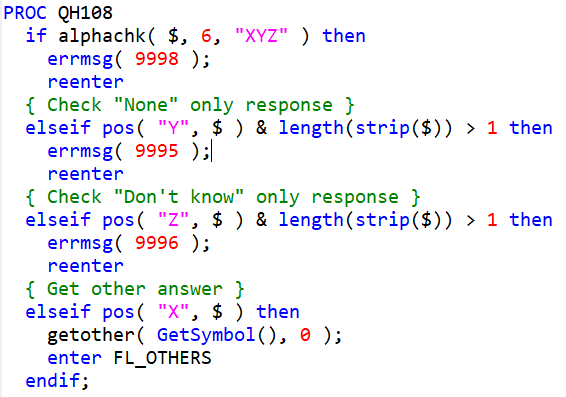
#### 4.3.2.12 Adding Logic to Check Land Area

In the Standard DHS-8 Household Questionnaire, Land Area question is only asked in one unit –hectares. Nonetheless, many surveys will have this standard question modified to include more than one type of unit to measure the land area. If this is the case, there is a call in the data capture program on the Land area variable to add code for checking the unit versus the amount of land area reported, and to take special attention to the amount of land area reported using the value 95, that means “95 or more hectares” in the standard DHS-8 Household questionnaire, which translates into “95 or more OF-THE-UNIT” when multiple units are used. Therefore, when using the value 95 it should refer to a unit of measurement, not to the special unit code ‘9’ used for special values like ‘Don’t know’ (9-98). For example, let us say that in a survey the units of land area are 1-hectares, 2-acres, and 9-special values. You will add code using a recode/endrecode statement to check valid values for each unit, including the unit for special values, and not allowing for a combination of 9-95. Otherwise, if the combination unit-number is not valid, you will enforce the input of a valid combination after informing the interviewer through an error message of the invalid combination entered.

PROC QH131  
{ !!! make sure to add code to properly check the units for different types of land }  
{     with the corresponding area.  Also make sure that if units are used, code 95  }  
{     should be accompanied to the unit it refers to. i.e., 9-95 isn't useful       }

#### 4.3.2.13 Adding Logic to New Survey Specific Variables

Go through all your survey specific variables and/or modules that you have identified while adjusting your survey dictionary to write the necessary logic needed to meet the criteria of the new variable and/or module. In general, filters for questions that are applicable only to a certain population are placed on the Preproc, for instance, questions for household members 15 to 24 years old or questions for women. Skips based on the value entered or validations of the value entered against other information previous entered or stored in an external file are placed on the Postproc. Onfocus logic are executed when the field becomes active, for example, you could place dynamically value sets on the Onfocus statement of the current field depending on the value entered in a prior field. Killfocus logic are executed when the current field stop being active, for instance, you could assign the value just entered in the current field to a working variable or to an array. When having survey specific alpha variables, do not forget to include a validation for the string entered using the DHS user defined function *alphachk*. In the example below, QH108 has six letters A to F, and a tail X, Y, and Z. Combinations of letters for multiple response are permitted except for letters Y (None) and Z (Don’t know). Whenever the letter X appears in the string, it should open the form FL\_OTHERS to type in the “other” response. Validations in QH108 are executed in the postproc even if the procedure QH108 does not specify it. When the logic does not specify what procedure the logic is in, the default is the postproc. When the logic includes more than one procedure (preproc, postproc, onfocus, killfocus) it is mandatory to write the procedure type to distinguish the order of execution.



For survey specific multiple-occurrence household member records, it is recommended to add an additional variable at the beginning of the record to store the column number, in addition to the household member line number, when the section or record is only applicable to a certain population, for example, household members 5 to 14 years old, household members that previously reported having accidentally injured, etc.

#### 4.3.2.14 Updating Filters and Skips When Changes Were Made to the Standard Questionnaires

During the questionnaire design visit, Survey Managers meet with the Technical Working Group (TWG) or Technical Committee (TC) to adapt the standard survey questionnaires into survey specific questionnaires. During this questionnaire adaptation process, Survey Managers can delete standard questions or add new survey specific ones. These changes may cause that standard skips and filters need to be updated accordingly. Usually, Survey Managers highlight these changes by color coding the questionnaires where changes were made. You will need to update the skips and filters that were altered as a result of the questionnaire adaptation process.

#### 4.3.2.15 Optional Modules to Delete If They Do Not Exist in your Survey

If your survey does not include the following modules, delete the code associated to the modules that do not exist in your survey from the Logic file:

Mosquito Nets

Child Labor

Child Discipline

Height, Weight, Hemoglobin (women)

Height, Weight, Hemoglobin (men)

## 4.4 Biomarker Questionnaires

### 4.4.1 Checking Second Decimal by Type of Scale

When weighting children 0-5 years old, women 15-49 years old, and men 15-54[59] years old in the biomarker questionnaire, we are now checking the second decimal obtained according to the type of scale. In the Postproc of PROC QB106/QB205/QB305, Weight in kilograms, there is a standard logic for scale type SECA 874. When using this scale, the second decimal should always be 0 or 5. The logic below displays a warning message if this is not the case. The function GetDecimal is a user defined function stored in DPCollect.apc in the \Library folder which returns the first or second digit of a decimal variable.

if !GetDecimal( $, 2 ) in 0,5 then  
  warning( 441 );  
endif;

0441 W Second decimal for weight different from 0,5

This logic needs to be modified if using scale type SECA 878 where the second decimal should be zero; or if using UNICEF scales that do not have second decimal at all, only one decimal. There is a note in the data entry application that you should also check that the second decimal is zero for UNICEF scales.

### 4.4.2 Checking Height is Bigger than Weight

There is a check in the Postproc of PROC QB108/QB207/QB307, Height in centimeters, to ensure that the height is bigger than the weight in kilograms, otherwise an error message is displayed demanding correction.

Biomarker for Children:

if QB108 in 20:140 & QB106 in 0:40 & QB108 < QB106 then  
  errmsg( 0451, "QB108", "QB106");  
  reenter  
endif;

Biomarker for Women:

if QB207 in 100.0:200.0 & QB205 in 20:150 & QB207 < QB205 then  
  errmsg( 0451, "QB207", "QB205");  
  reenter  
endif;

Biomarker for Men:

if QB307 in 130.0:210.0 & QB305 in 40:170 & QB307 < QB305 then  
  errmsg( 0451, "QB307", "QB305");  
  reenter  
endif;

0451 E Height "%s" smaller than weight "%s"; they should possibly be reversed!

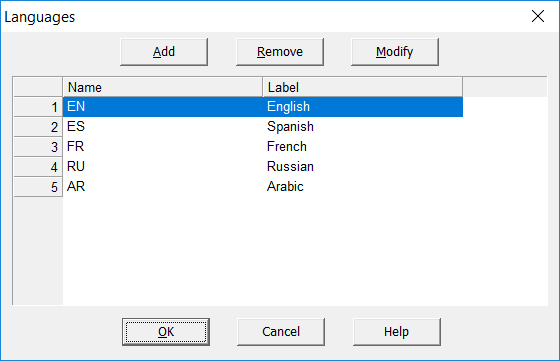
### 4.4.3 Double Entering Measurements for Weight, Height, and Hemoglobin

Measurement values for Weight, Height, and Hemoglobin are double entered as a data quality control to avoid typing errors. When double entering these measurement values, the first recorded measurement becomes hidden to avoid interviewers from copying the values seen at the tablet’s screen. If the two entries differ, an error message is displayed, and the cursor is returned to the first measurement field to reenter the measurement values twice again. The field to reenter weights is QB106W/QB205W/QB305W; for height it is QB108W/QB207W/QB307W; and for hemoglobin it is QB122W/QB225W/QB325W.

## 4.5 QSF

Questionnaires in any given CAPI survey can be available in multiple languages that are commonly spoken in the country. For example, in Ghana surveys, questionnaires are normally available in Akan, Ga, and Ewe local languages besides English. These languages are stored in the Question File or QSF file to allow interviewers to select the language used during the interview in the CAPI program. The Data Processor is responsible to update the Reference language in the CAPI program, that could be English, French, Spanish, Russian, or Arabic and will work with local counterparts who will add translations of the question texts into the other languages used in the survey. Usually, the Survey Manager will provide the DP the counterpart persons’ names that will collaborate in the translations of the different languages used in the survey. For example, if there are three local languages in a DHS survey, there may be three different persons responsible for translating into each of these languages because normally local languages are spoken regionally and need a representor from the region that is able to speak and write in the local language.

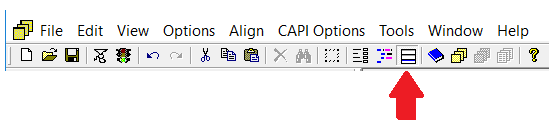
In the CAPI data entry application, the Data Processor first need to define the languages using the CAPI Options menu and then Define CAPI Languages. In the window that appears, use the Remove button to remove languages that will not be used in the survey and then use the Add button to add the languages used in the survey by specifying a two-digits language name and the language itself as the label.



When the Data Processor works with the QSF in the Reference language (English, French, Spanish, Russian, or Arabic), s/he will use the paper questionnaire provided by the Survey Manager and look question by question to search for modifications to the DHS standard text and to update the changes in the Reference language (English, French, Spanish, Russian, or Arabic) in the QSF. It is expected that the Survey Manager use color coding to mark changes made in the questionnaire, during the design visit, including question texts. This facilitate a more effective work for the Data Processor and local counterparts who will also work with the QSF.

When working with the QSF, there are conventions used to instruct the interviewers while conducting the survey. Text in black color are to be read to the respondent. Text in blue color with all capital letters are instruction for interviewers. These instructions to the interviewer should remain in the Reference language (English, French, Spanish, Russian, or Arabic) and not be translated into local languages. Only the question text should be translated.

The icon to access the CAPI Question in CSPro is:

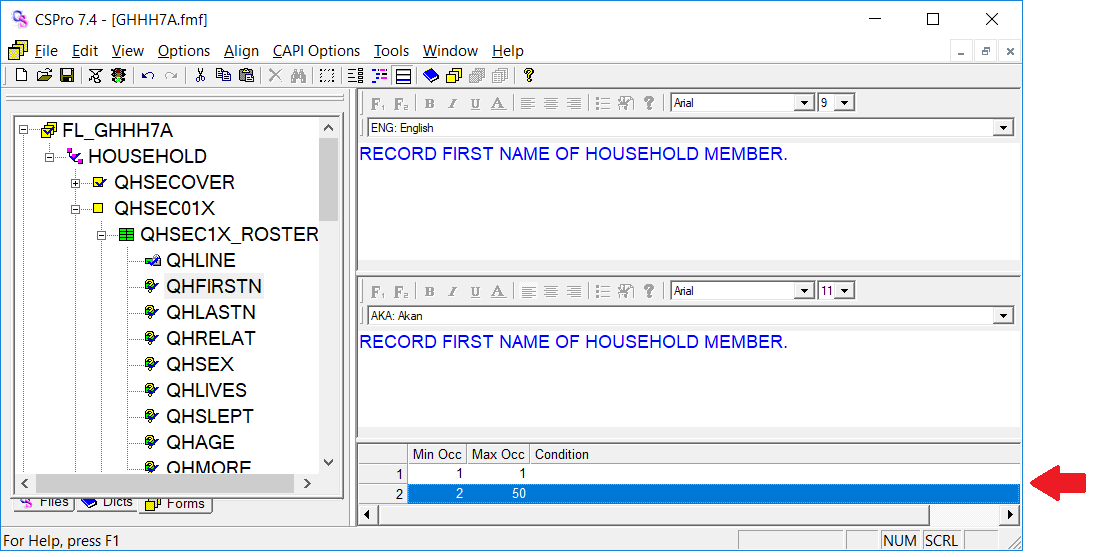


Most questions or variables will have a correspondent one-to-one question per variable in the QSF. But for some variables in multiple occurrence records, there is a need to ask the question differently for the first occurrence compared to the rest of the occurrences in a roster. In these cases, the Data Processor will define more than one question text per occurrence in CSPro in the pertaining variable. For example, the first occurrence of variable QHFIRSTN will have a question in black to be read to the respondent and an instruction in blue with all capital letters for the interviewer. Subsequent occurrences from 2-50 will only have the instruction for the interviewer, without the need to read the question again to the respondent.

First occurrence of QHFIRSTN:

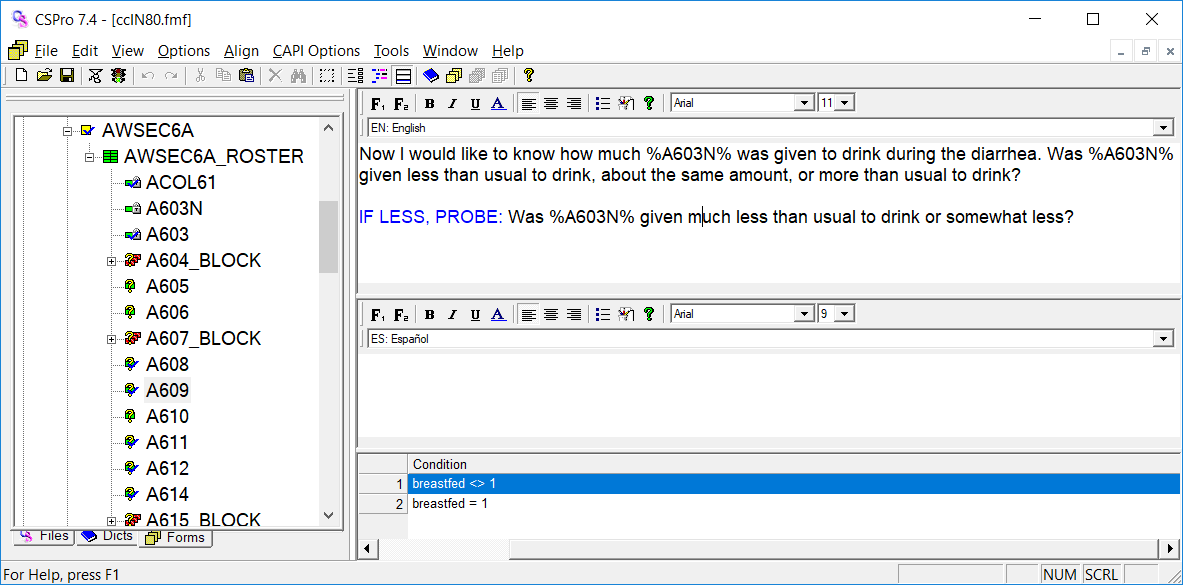
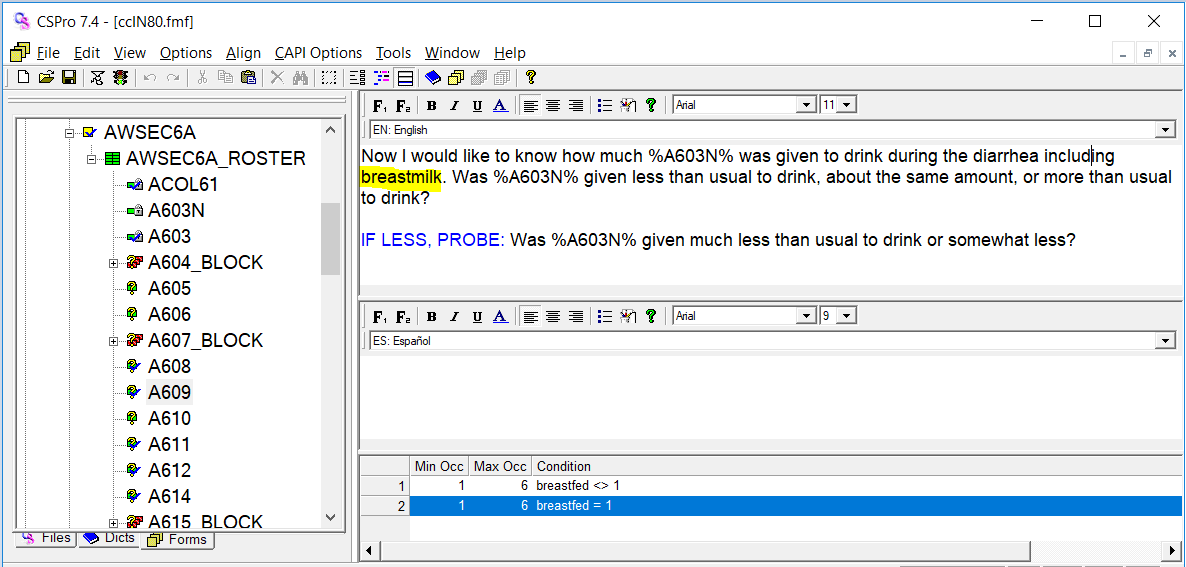


Occurrences 2-50 of QHFIRSTN:



Question text can be customized using fills, or text substitutions. These fills reference the contents of variables, dictionary items, or user-defined functions. They are identified by the name of the object surrounded by percent characters (%). During the interview when the data entry application is running, the value of the variable is substituted into the question text. For example, " Is %QHFIRSTN% %QHLASTN% male or female?" might be transformed into "Is Mary Smith male or female?" Fills help customize the question text and frame it for specific respondents.

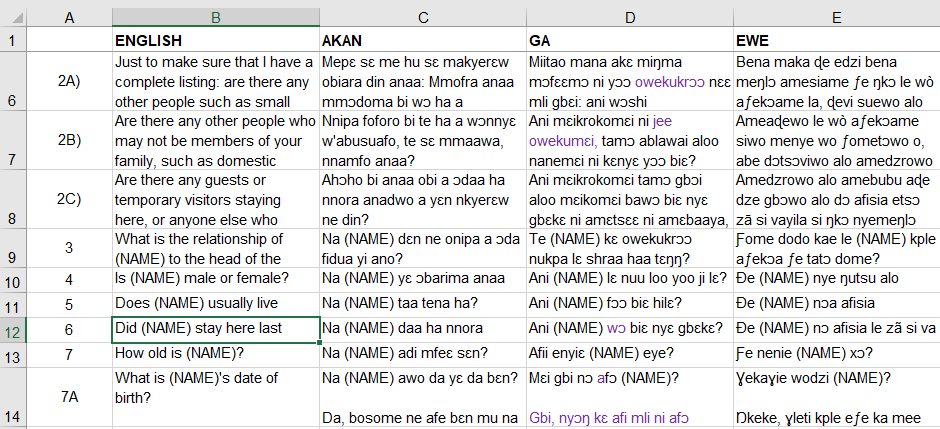
Data Processors can also define conditions for questions text. For example, in the data entry program for individual questionnaire CCIN80.apc, question Q609 (Drinking pattern with diarrhea) is asked differently if the child is being breastfed, and this is done by adding the condition breastfed <> 1 and breastfed = 1 in two different lines (line 1 and line 2) for all occurrences in this roster (which maximum is six occurrences).



### 4.5.1 QSF Editor

After the QSF files for the Household (CollectHH.qsf) and the Individual (CollectIN.qsf) are completed in the Reference language (English, French, Spanish, Russian, or Arabic), the Data Processor will provide them to the local counterparts who will add the translations into the local languages. The Data Processor will also provide to the counterparts the Excel survey questionnaires that have the translations sheet at the end of the file, which contains translations in different languages for each question. These translations will be used by the counterpart staff to copy and paste them into the QSF Editor and add some other edits. The Data Processor and the local counterpart staff should follow the “Instructions for QSF Editor” manual located in SharePoint -> Standard-7 DHS -> Data Processing -> CC71 -> CAPI Documentation -> CAPI manuals -> Instructions for QSF Editor.doc. This is a very detailed manual that explains what the local counterpart is expected to do with the QSF file. The Data Processor should get familiar with this manual to further explain the steps on how to add local languages in the QSF file to local counterpart staff when needed.

Example of a translation sheet for the Household questionnaire



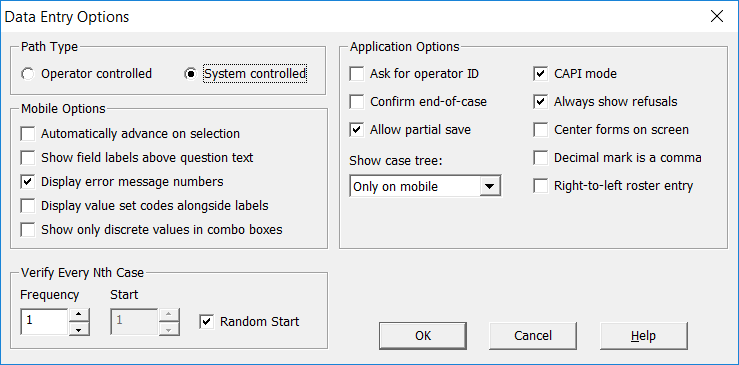
### 4.5.2 Questionnaire Extraction Tools

The QSF Editor is a convenient tool to add local languages into the QSF file, nonetheless, it is not automated and requires that local counterpart staff copy and paste translated text from the Excel questionnaire translation sheet into the QSF Editor. To solve this, in 2019 Keith Purvis developed four tools called Questionnaire Extraction Tools that allows to create a QSF directly from the Excel questionnaire file. Some tweaking is necessary. A 13 pages documentation exist explaining how to create the QSF using the Questionnaire Extraction Tools. This documentation and the necessary programs for the Questionnaire Extraction Tools are in SharePoint -> Teams & Topics -> Data Processing -> In the left menu, select “QSF and CAPI utilities”. (This topic could be more detailed in the future when these tools are frequently used)

## 4.6 Options Menu

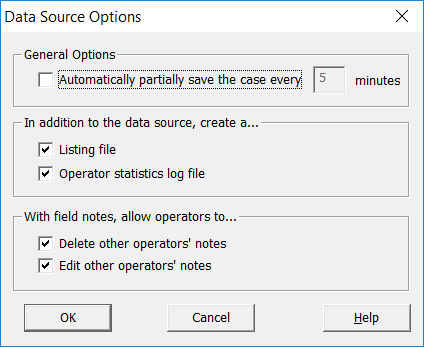
### 4.6.1 Data Entry Option

The image below shows the Data Entry Option. First, the path type in the data entry form is System controlled as opposed to Operator controlled. Under Mobile Options, the only option ticked is Display error message numbers whenever an error or warning message is displayed. This is useful because most messages in the DHS CAPI programs are referenced by the message number in the Supervisor’s manual or the Secondary Editing manual with instructions on how to correct them. Another option to highlight under Mobile Options is Display value set codes alongside labels, this is the default in Windows tablets, but not in Android tablets, so you may have to tick it if using Android tablets. In version 7.4, even when ticking this option, codes still were not shown alongside labels. Under Application Options, it is important to have “Allow partial save”, “CAPI mode”, and “Always show refusals” ticked. When countries have comma as decimal marks or write from right to left, you might tick those options under Application Options.



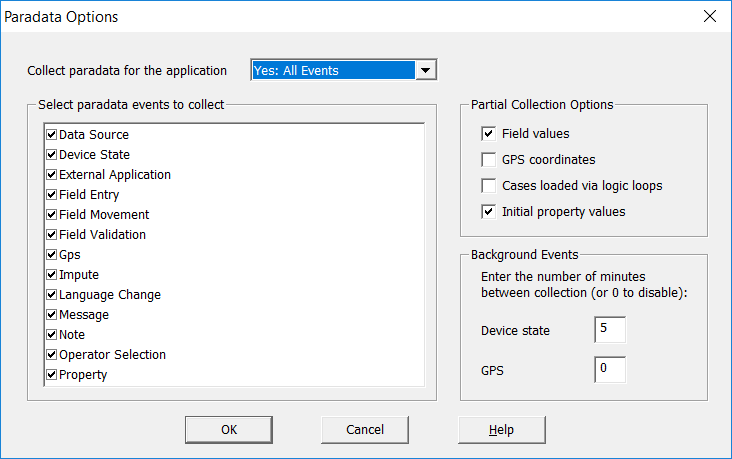
### 4.6.2 Data Sources Option

Using the Data Sources Options, we can specify CSPro to create additional Listing and Operator statistics log files. Also, we can indicate permissions to the Notes records, this is allow to delete or edit other interviewers’ notes.



### 4.6.3 Paradata Options

The Paradata Options in the CollectHH application collects All paradata events. Under Partial Collection Options, Field and Initial property values are marked. Under Background Events, the number of minutes between collection for the Device is set to 5 minutes whereas the number of minutes for the GPS is disable.



## 4.7 Testing your Programs

Do not assume that a logic that is compiling is working properly or as intended. You must always test your programs to verify that any updated or new logic added to the program is working the way it is intended. Especially logic added to new survey specific variables or new survey specific modules or adjusted logic from the standard programs must be tested. Logic that remain the same from the standard DHS programs has been thoroughly tested over the many years of The DHS Program, unless there are new variables/logic introduced as part of the Revision process of a new phase of DHS questionnaires, at the beginning of the phase there may be small errors in the new logic introduced for the new variables or modules added. Thus, it is absolutely a must to test your programs thoroughly. To test your programs, run your applications from the menu (Interviewer, Supervisor, or Central Office menus) and pretend to be interviewing. To help you enter ‘real’ data into the questions, think about your own information or information from a family member or a friend. This information is for testing purpose, after applications have been thoroughly tested and you are comfortable with the results, please delete any information entered especially if they contain PII. Once the pretest or main training start, you will have plenty of information from the training to continue debugging further applications (for example, Central Office programs).