



CONDUCTING TABLET-BASED FIELD DATA COLLECTION WITH CSPRO

A Handbook

FEBRUARY 2020



Food and Agriculture
Organization of the
United Nations



CONDUCTING TABLET-BASED FIELD DATA COLLECTION WITH CSPRO

A Handbook

February 2020

Food and Agriculture Organization of the United Nations

and

Asian Development Bank

Bangkok, 2020

FAO and ADB. 2020. *Conducting Tablet-Based Field Data Collection with CSPro: A Handbook*. Bangkok.
<https://doi.org/10.4060/ca7690en>

The designations employed and the presentation of material in this information product do not imply the expression of any opinion whatsoever on the part of the Food and Agriculture Organization of the United Nations (FAO) or Asian Development Bank (ADB) concerning the legal or development status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. The mention of specific companies or products of manufacturers, whether or not these have been patented, does not imply that these have been endorsed or recommended by FAO or ADB in preference to others of a similar nature that are not mentioned.

The views expressed in this information product are those of the author(s) and do not necessarily reflect the views or policies of FAO or ADB.

ISBN 978-92-5-132179-9 [FAO]
© FAO and ADB, 2020



Some rights reserved. This work is made available under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 IGO licence (CC BY-NC-SA 3.0 IGO; <https://creativecommons.org/licenses/by-nc-sa/3.0/igo/legalcode>).

Under the terms of this licence, this work may be copied, redistributed and adapted for non-commercial purposes, provided that the work is appropriately cited. In any use of this work, there should be no suggestion that either FAO or ADB endorses any specific organization, products or services. The use of the respective logos of FAO or ADB is not permitted. If the work is adapted, then it must be licensed under the same or equivalent Creative Commons license. If a translation of this work is created, it must include the following disclaimer along with the required citation: "This translation was not created by either the Food and Agriculture Organization of the United Nations (FAO) or the Asian Development Bank (ADB). Neither FAO nor ADB is responsible for the content or accuracy of this translation. The original English edition shall be the authoritative edition."

Disputes arising under the licence that cannot be settled amicably will be resolved by mediation and arbitration as described in Article 8 of the licence except as otherwise provided herein. The applicable mediation rules will be the mediation rules of the World Intellectual Property Organization <http://www.wipo.int/amc/en/> mediation/rules and any arbitration will be in accordance with the Arbitration Rules of the United Nations Commission on International Trade Law (UNCITRAL)

Third-party materials. Users wishing to reuse material from this work that is attributed to a third party, such as tables, figures or images, are responsible for determining whether permission is needed for that reuse and for obtaining permission from the copyright holder. The risk of claims resulting from infringement of any third-party-owned component in the work rests solely with the user.

Sales, rights and licensing. FAO information products are available on the FAO website (www.fao.org/publications) and can be purchased through publications-sales@fao.org. Requests for commercial use should be submitted via: www.fao.org/contact-us/licence-request. Queries regarding rights and licensing should be submitted to: copyright@fao.org.

ADB information products are available on the ADB website (www.adb.org/publications).

Contents

Foreword	xi
Abbreviations.....	xii
Chapter 1: Introduction to CAPI	1
1.1 What is CAPI?	1
1.2 Why use CAPI?	1
1.3 Hardware requirements	2
1.3.1 Tablets	2
1.3.2 Power Solutions	2
1.3.3 Internet Connection	2
1.3.4 Personal Computers and Laptops.....	2
1.3.5 Server.....	3
1.4 CAPI Project Workflow	3
Chapter 2: Introduction to CSPro	5
2.1 Data Flows in a CAPI Project	5
2.2 Installing CSPro on a Personal Computer.....	6
2.3 Installing CSEntry on a Tablet	7
2.4 Data Storage Options.....	7
2.4.1 Dropbox	7
2.4.2CSWeb	7
2.4.3File Transfer Protocol Server.....	7
2.5 CAPI Question Types	7
2.5.1 Numeric Questions.....	7
2.5.2Text Questions.....	8
2.5.3Single Select Questions.....	8
2.5.4Multi Select Questions.....	8
2.5.5Date Questions	8
2.6 Creating a CAPI Questionnaire in CSPro	9
Chapter 3: Starting a New Project.....	10
Chapter 4: Setting Up the Data Dictionary	12
4.1 Identification Items	12
4.2 Levels	14
4.3 Records	14
4.4 Rosters	15
4.5 Items	16
4.5.1Text Questions.....	17
4.5.2 Numeric Questions.....	17
4.5.3Single Select Questions.....	17
4.5.4Multi Select Questions.....	19
4.5.5Date Questions	20

Chapter 5: Creating Forms and Question Text	21
5.1 Creating Forms for CAPI.....	21
5.2 Blocks	22
5.3 Adding New Questions.....	24
5.4 Creating Question Text and Interviewer Instructions	25
5.4.1 Setting Help Text.....	26
5.4.2 Dynamic Text Display in Roster Questions.....	26
5.4.3 Question Text Macros	27
5.5 Displaying the Calendar for Date Questions.....	27
Chapter 6: Skips and Validations	29
6.1 An Overview of Skips	29
6.2 Logic Structure	29
6.2.1 Global vs Procedural.....	29
6.2.2 Procedural Types	29
6.3 Basic Syntax for Implementing Skips.....	30
6.4 Other Useful Movement Commands	32
6.5 Relational and Logical Operators	33
6.6 Multi Select Skips.....	34
6.7 Skips for Rosters	34
6.7.1 Setting Up Occurrence Labels.....	34
6.7.2 Ending a Nonfixed Roster.....	36
6.7.3 Skips Based on Answers in a Previous Roster.....	40
6.7.4 Questions for Particular Occurrences.....	40
6.8 Skip to Next	40
6.9 An Overview of Validations.....	41
6.10 Hard and Soft Validations.....	41
6.11 Dynamic Text in Validation Error Messages.....	42
6.12 Date Validations.....	42
6.13 Multiple Value Sets	43
Chapter 7: Surveys Conducted in Multiple Languages.....	46
7.1 Alternative Languages in the Data Dictionary	46
7.2 Alternative Languages in Question Text.....	47
7.3 Alternative Languages in Error Messages	50
Chapter 8: Setting Up the Completed Questionnaire for Data Collection.....	51
8.1 Setting the Syncing Protocols in CSPro	51
8.2 Refining the Data Collection Options in CSPro.....	52
8.3 Installing CSEntry onto Tablets.....	52
8.4 Deploying the Questionnaire to the Tablets	55
8.4.1 Deploying Directly to the Tablet	55
8.4.2 Deploying via Dropbox.....	56
8.5 Questionnaire Updates via the Server	57

Chapter 9: Quality Control during Data Collection	59
9.1 Downloading Data from the Server	59
9.2 Making and Viewing Survey Notes	60
9.3 Quality Control Tables	61
9.4 Batch Editing for Error Checks.....	62
9.5 Paradata	66
Chapter 10: Advanced Features	69
10.1 Timestamps	69
10.2 Capturing Location Coordinates	69
10.3 Accessing Other Apps on the Tablet	71
10.4 Lookup Tables.....	72
10.4.1 Creating the Data Dictionary for a Lookup Table	72
10.4.2.Formatting the Lookup File for CSPro	74
10.4.3 Writing Logic Code for the Lookup Table	74
10.5 Task Menus.....	75
10.6 Bluetooth Syncing	76
10.7 Reports.....	78
Chapter 11: Help and Support.....	81
11.1 Help Documentation	81
11.2 Online Users Forum.....	82
11.3 Example Projects.....	82
References	83

Tables

Table 1.1: Tablet Specifications for CSEntry, June 2019	2
Table 1.2: Hardware Specifications for CSPro.....	2
Table 6.1: Relational Operators Used in CSPro	33
Table 6.2: Logical Operators Used in CSPro.....	34

Figures

Figure 2.1: Data Flows in a Simple CAPI Project using CSPro	5
Figure 2.2: Downloading CSPro for Installation on Personal Computer	6
Figure 2.3: Information Form to Download CSPro	6
Figure 2.4: Installing CSEntry from the Google Play Store	7
Figure 2.5: Example of a Numeric Question in CSEntry	8
Figure 2.6: Example of a Text Question in CSEntry	8
Figure 2.7: Example of a Single Select Question in CSEntry.....	8
Figure 2.8: Example of a Multi Select question in CSEntry.....	9
Figure 2.9: Example of a Date Question in CSEntry.....	9
Figure 3.1: CSPro Desktop Icon	10
Figure 3.2: Creating a New Project in CSPro	10
Figure 3.3: Creating a CAPI Survey in CSPro	10
Figure 3.4: Selecting the Directory and Naming the CAPI Survey	10

Figure 3.5:	Creating a Blank Data Dictionary	11
Figure 3.6:	Naming the New Data Dictionary File.....	11
Figure 4.1:	Tree View in the Data Dictionary of CSPro	12
Figure 4.2:	Identification Item in Tree and Item View	13
Figure 4.3:	Modifying an Existing Identification Item	14
Figure 4.4:	Multiple Identification Items.....	14
Figure 4.5:	Adding Records to the Questionnaire	15
Figure 4.6:	Simple Representation of Rostered Questions.....	15
Figure 4.7:	Setting the Maximum Number of Roster Repeats	15
Figure 4.8:	Setting Occurrence Labels for a Rostered Record.....	16
Figure 4.9:	Example of Completed Occurrence Labels	16
Figure 4.10:	Adding an Item to a Record.....	17
Figure 4.11:	Adding an Item for a Text Question	17
Figure 4.12:	Adding an Item for a Numeric Question	17
Figure 4.13:	Adding an Item for a Single Select Question	18
Figure 4.14:	Adding a Value Set for a Single Select Question	18
Figure 4.15:	Example of a Value Set for a Single Select Question.....	19
Figure 4.16:	Modifying an Existing Value Set	19
Figure 4.17:	Adding an Item for a Multi Select Question	19
Figure 4.18:	Example of a Value Set for a Multi Select Question.....	20
Figure 4.19:	Adding an Item for a Date Question.....	20
Figure 5.1:	Generating Forms for a CAPI Project	21
Figure 5.2:	Options Dialogue Box when Creating Forms	22
Figure 5.3:	Navigating between Files, Dictionaries, and Forms.....	22
Figure 5.4:	Tree View of Forms and Listing of Items	23
Figure 5.5:	Adding Blocks to Forms.....	23
Figure 5.6:	Example of Questions Placed into a Block.....	23
Figure 5.7:	Adding a New Question to the Data Dictionary.....	24
Figure 5.8:	Dragging a Newly Added Question from the Dictionary to the Forms	24
Figure 5.9:	Default Order of Newly Added Dictionary Item on Form.....	25
Figure 5.10:	Corrected Order of Newly Added Dictionary Item on Form.....	25
Figure 5.11:	Viewing and Editing Question Text.....	25
Figure 5.12:	Setting the Font of the Question Text	26
Figure 5.13:	Adding Help Text to a Question	26
Figure 5.14:	Dynamically Displaying the Occurrence Label	26
Figure 5.15:	Accessing Question Text Macros	27
Figure 5.16:	Dialogue Box for Question Text Macros	27
Figure 5.17:	Applying Field Properties to a Date Question.....	28
Figure 5.18:	Modifying the Capture Type for a Date Question	28
Figure 6.1:	Navigating to the System Logic in CSPro	29
Figure 6.2:	Accessing the Global Logic Section	30
Figure 6.3:	Example of a Procedural Logic Section	30
Figure 6.4:	Basic Logic Commands for Skips	31
Figure 6.5:	Using an “If” Statement in Logic Code	31

Figure 6.6:	Example of Using an “If” Statement to Create a Skip	31
Figure 6.7:	Example of Using an “Ask If” Command to Create a Skip	31
Figure 6.8:	Using the Compile Button to Check for Errors.....	32
Figure 6.9:	Example of a Compilation Error in Logic Code	32
Figure 6.10:	Example of Successfully Compiled Logic Code.....	33
Figure 6.11:	Creating Code to Search for Alpha Characters.....	34
Figure 6.12:	Example of Code for an Alpha Character Search.....	34
Figure 6.13:	Setting Occurrence Labels for a Fixed Roster.....	35
Figure 6.14:	Example of Occurrence Labels in a Fixed Roster.....	35
Figure 6.15:	Dynamically Determining Occurrence Labels for Nonfixed Rosters	36
Figure 6.16:	Finding the Roster Name in the Form Viewer	36
Figure 6.17:	Example of Code for Occurrence Labels in a Nonfixed Roster.....	36
Figure 6.18:	Using a Controlling Question to End a Nonfixed Roster	37
Figure 6.19:	Adding a Controlling Question to the Form	37
Figure 6.20:	Adjusting the Order of a Controlling Question.....	38
Figure 6.21:	Setting Roster Properties for a Controlling Question.....	38
Figure 6.22:	Setting the Occurrence Control Field	38
Figure 6.23:	Checking after Each Occurrence to End a Nonfixed Roster.....	39
Figure 6.24:	Adding an Additional Question to End a Nonfixed Roster.....	39
Figure 6.25:	Example of Logic Code to End a Nonfixed Roster	40
Figure 6.26:	Basic Syntax to Check for a Condition in a Previous Roster.....	40
Figure 6.27:	Example of Code Using the Count Function	40
Figure 6.28:	Example of Code to Ask Questions by Occurrence.....	40
Figure 6.29:	Basic Syntax for a Hard Validation	41
Figure 6.30:	Example of Code for a Hard Validation	42
Figure 6.31:	Basic Syntax for a Soft Validation.....	42
Figure 6.32:	Example of Code for a Soft Validation	42
Figure 6.33:	Displaying Previous Answers in Error Messages	42
Figure 6.34:	Inserting Values for Previous Answers.....	42
Figure 6.35:	Setting the Date Format for an Age Validation	43
Figure 6.36:	Basic Syntax to Calculate the Difference between Dates	43
Figure 6.37:	Example of Regional Identifiers for Use in Multiple Value Sets.....	44
Figure 6.38:	Example of Multiple Value Sets for District	44
Figure 6.39:	Example of Code to Display Answer Options by Value Set.....	45
Figure 7.1:	Example of Screen Areas to Display Additional Languages	46
Figure 7.2:	Accessing the Language Options in the Data Dictionary	47
Figure 7.3:	Adding a New Language in the Data Dictionary	47
Figure 7.4:	Selecting an Alternative Language for the Questionnaire	47
Figure 7.5:	Adding the Alternative Language to Item Labels.....	48
Figure 7.6:	Adding the Alternative Language to Value Sets.....	48
Figure 7.7:	Adding the Alternative Language to Occurrence Labels	48
Figure 7.8:	Defining Alternative Languages for Question Text	49
Figure 7.9:	Adding the Alternative Language for Question Text.....	49
Figure 7.10:	Example of Text Piping within Alternative Language Text.....	49

Figure 7.11:	Basic Syntax to Display an Alternative Language in an Error Message	50
Figure 7.12:	Example of an Error Message Displayed on a Tablet	50
Figure 8.1:	Accessing the Syncing Options in CSPro.....	51
Figure 8.2:	Example of Signing In to a Server (Dropbox)	51
Figure 8.3:	Establishing the Syncing Protocols.....	52
Figure 8.4:	Accessing the Data Entry Options in CSPro	52
Figure 8.5:	Allowing for Saving of Partially Completed Questionnaires.....	53
Figure 8.6:	Determining the CSPro Version in Use	53
Figure 8.7:	Downloading the CSEntry Install File	53
Figure 8.8:	Copying the CSEntry Install File to the Tablet	54
Figure 8.9:	Allowing Installation of Apps from Unknown Sources	54
Figure 8.10:	Accessing the Install File on the Tablet	54
Figure 8.11:	Initiating the Install File via Device Storage	55
Figure 8.12:	CSEntry Icon on Tablet	55
Figure 8.13:	Initiating Deployment of the Questionnaire from CSPro.....	55
Figure 8.14:	Specifying the Deployment Options in CSPro	56
Figure 8.15:	Creating the Destination Folder for the Deployment Files.....	56
Figure 8.16:	Questionnaire Files Deployed to Tablet	56
Figure 8.17:	Preparing to Import the Questionnaire into the Tablet.....	57
Figure 8.18:	Selecting the Source to Import the Questionnaire.....	57
Figure 8.19:	Installing the Questionnaire onto the Tablet	57
Figure 8.20:	Questionnaire Ready for Use on Tablet	57
Figure 8.21:	Syncing Options for Connecting to the Server (Dropbox)	57
Figure 8.22:	Overwriting the Questionnaire Files on the Server (Dropbox)	58
Figure 9.1:	Opening the Data Viewer in CSPro	59
Figure 9.2:	Initiating the Download of a Data File.....	59
Figure 9.3:	Specifying the Server to Retrieve the Data File	59
Figure 9.4:	Downloading the Data File to the Local Directory	60
Figure 9.5:	Accessing the Field Note Functionality in CSEntry.....	60
Figure 9.6:	Entering a Field Note on the Current Question.....	60
Figure 9.7:	Converting the Data File to a Text File	60
Figure 9.8:	Saving the Text File to the Local Directory	61
Figure 9.9:	Example of Field Notes Displayed in CSPro Text Viewer.....	61
Figure 9.10:	Creating Quality Control Tables in CSPro	62
Figure 9.11:	Selecting a Data Dictionary for Tabulation.....	62
Figure 9.12:	Dragging Variables from the Data Dictionary to the Tables	63
Figure 9.13:	Example of a Quality Control Table in CSPro	63
Figure 9.14:	Creating a New Batch Editing Process	63
Figure 9.15:	Saving a New Batch Editing Process	64
Figure 9.16:	Specifying the Data Dictionary for Batch Editing.....	64
Figure 9.17:	Screen Layout Used for Batch Editing	65
Figure 9.18:	Example of Logic Code for Batch Editing.....	65
Figure 9.19:	Initiating the Batch Editing Process.....	65
Figure 9.20:	Example of Report Generated after Batch Editing	66

Figure 9.21: Accessing Paradata Options in CSPro	66
Figure 9.22: Setting the Events for Paradata Collection	67
Figure 9.23: Accessing the Paradata File on the Tablet.....	67
Figure 9.24: Example of Paradata Displayed in CSEntry.....	67
Figure 9.25: Combining Paradata Files from Multiple Tablets	68
Figure 10.1: Example of Basic Syntax to Capture a Timestamp	69
Figure 10.2: Setting Latitude and Longitude Variables.....	70
Figure 10.3: Logic Code to Capture Latitude and Longitude	70
Figure 10.4: Basic Syntax to Set Up the Userbar Icon	70
Figure 10.5: Userbar Icon in CSEntry	70
Figure 10.6: Accessing the Menu Options in the Userbar	71
Figure 10.7: Basic Syntax to Access Other Apps.....	71
Figure 10.8: Example of Logic Code for Taking a Photo	71
Figure 10.9: External File Used for a Lookup Table (Excel).....	72
Figure 10.10: Creating a New Data Dictionary for a Lookup Table	72
Figure 10.11: Defining the Variables for a Lookup Table.....	73
Figure 10.12: Structure of Completed Data Dictionary for a Lookup Table	73
Figure 10.13: Adding the Data Dictionary for the Lookup Table.....	73
Figure 10.14: Setting an External Dictionary for a Lookup Table	73
Figure 10.15: Converting the Excel Lookup File to CSPro.....	74
Figure 10.16: Selecting the External Data File for Lookup	74
Figure 10.17: Basic Syntax to Scan a Lookup File	74
Figure 10.18: Example of Logic Code to Utilize a Lookup Table.....	75
Figure 10.19: Example of a Task Menu in CSPro.....	75
Figure 10.20: Example of Syntax to Launch a Task from a Menu.....	76
Figure 10.21: Standard Syncing Setup with Bluetooth for Backup	76
Figure 10.22: Syncing Setup with Bluetooth Connection to Supervisor	77
Figure 10.23: Example of Logic Code for Bluetooth Syncing	77
Figure 10.24: Example of Bluetooth Syncing on CSEntry	78
Figure 10.25: Basic Syntax to Open a Report.....	78
Figure 10.26: Running the Task to Generate a Report.....	78
Figure 10.27: Selecting the External Text File to Generate a Report.....	78
Figure 10.28: Example of Logic Code to Insert Text into a Report	79
Figure 10.29: Example of Logic Code to Define Numeric Variables for a Report	79
Figure 10.30: Logic Code to Count Variable Frequencies	79
Figure 10.31: Logic Code to Insert Counts into a Report	79
Figure 10.32: Example of Logic Code to View a Report.....	79
Figure 10.33: Example of a Report Displayed on a Tablet.....	80
Figure 11.1: Accessing Help Topics within CSPro.....	81
Figure 11.2: Help Functions and Documentation within CSPro	81
Figure 11.3: CSPro Users Forum	82
Figure 11.4: Accessing Project Examples Created by CSPro	82
Figure 11.5: Opening Project Examples from the Systems Directory	83

Foreword

The Asian Development Bank (ADB) and the Food and Agriculture Organization (FAO) of the United Nations are pleased to present this publication, *Conducting Tablet-Based Field Data Collection with CSPro: A Handbook*.

ADB and the FAO strongly support the improvement of national statistical systems by building the technological capacities of national statistical offices and line ministries. Recognizing the substantial data requirements for monitoring the Sustainable Development Goals, both organizations acknowledge the need to assist producers of official statistics and actively encourage the adoption of innovative technologies that support timely and effective monitoring of the Goals.

Over the years, data collection using handheld digital devices—often referred to as computer-assisted personal interviewing (CAPI)—has gained popularity in survey research. This is largely due to its potential to improve data quality and provide quicker turnaround on results compared to traditional data collection methods. CAPI also offers scope for new question types that enable the collection of geospatial and multimedia data, including global positioning system coordinates and audio, photographic, and video files.

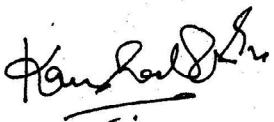
Because the capacity to adopt CAPI has been limited in many economies across the Asia and Pacific region, ADB and the FAO developed two massive online open courses (MOOCs) on CAPI, targeted at countries in the region (and around the world) more quickly and cost-efficiently. The online courses featured two freely available CAPI software platforms being used by official data collection agencies to conduct censuses and surveys—Census and Survey Processing (CSPro) Android and Survey Solutions.

This handbook has been developed to complement the MOOC on CSPro. It is suitable for anyone who has experience in conducting traditional face-to-face interviews and wishes to learn the workflows and skills needed to conduct a CAPI project. The handbook's instructions are tailored toward people with beginner or intermediate experience in CSPro, particularly if they are looking to reinforce existing knowledge or learn additional functionalities.

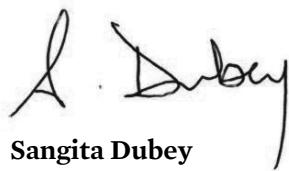
The handbook was originally written in November 2018, based on the most recent version of CSPro at that time (Version 7.1). It was revised in May 2019 to reflect additional functionalities of the current software (Version 7.2).

We would like to thank those who contributed to the production of this publication for their dedication and hard work. The ADB team was supervised by Kaushal Joshi, and this publication was led by Lakshman Nagraj Rao. The main body of the text was drafted by Lachlan Bruce, with significant inputs from Pamela Lapitan, Anna Christine Durante, Dave Pilon, Guido Pieraccini, and Jude David Roque. The FAO team worked under the supervision of Sangita Dubey of the FAO Regional Office for Asia and the Pacific and included Anthony Burgard and Sanghyun Jeon. We also acknowledge the contribution of Paul Dent as the manuscript editor and Rhommell Rico as the focal person for the publication's design, layout, and typesetting. We would also like to thank the International Programs of the United States Census Bureau for providing us valuable advise during the preparation of this handbook and to USAID for financing the development of a free CAPI platform that has revolutionized survey data collection and management across the world.

This handbook is designed to help national statistical officers and other interested readers embrace the efficiencies of CAPI-based data collection to supersede the traditional pen and paper interviewing method. We hope that it contributes to the adoption of other innovative tools and technologies that further strengthen national statistical systems.



Kaushal Joshi
Principal Statistician
Economic Research and Regional Cooperation Department
Asian Development Bank



Sangita Dubey
Regional Statistician
FAO Regional Office for Asia-Pacific

Abbreviations

ADB	Asian Development Bank
apk	Android package
CAPI	computer-assisted personal interviewing
CSPro	Census and Survey Processing
FTP	file transfer protocol
GPS	global positioning system
ID	identification
IT	information technology
PAPI	pen and paper interviewing
PC	personal computer
PSU	primary sampling unit
RAM	random access memory
SIM	subscriber identification module
SPSS	Statistical Package for Social Sciences
USB	universal serial bus
USCB	United States Census Bureau

Chapter 1: Introduction to CAPI

1.1 What is CAPI?

Computer-assisted personal interviewing (CAPI) is a viable alternative to paper-based surveying methods, or pen and paper interviewing (PAPI). As the name suggests, the key difference between CAPI and paper-based methods is that computers or handheld devices are used to display the questions to be asked by interviewers and to record the answers of the respondents.

CAPI has been used since the early 1990s, often utilizing laptop computers or older mobile technology such as Blackberry devices and personal digital assistants such as PalmPilots. In recent years, improvements in CAPI software and mobile devices has increased the use of the technique all around the world.

CAPI has emerged as a preferred option because, where surveys are conducted using tablets or smart phones, the need to digitize data is eliminated and the quality of the collection method is improved (given the checks and balances that can be built into an automated system). It generally means that datasets are ready for analysis, and for interpretation by policymakers, much faster than they might be using traditional methods.

1.2 Why Use CAPI?

CAPI has several advantages over PAPI:

Data quality. CAPI eliminates the need to digitize data, while improving quality through a series of built-in checks. CAPI has the ability to validate data in real time because the platform's programming can allow for automated skip patterns, display error messages whenever unexpected values

are entered by the interviewer, and follow other validation rules (e.g., ranges of values).

Data security. Following interviews, data can be instantly uploaded to a “cloud” server or a physical server. This means the datasets are secured almost instantly, mitigating the risk of losing data by having to transfer paper forms to a central location for data entry (e.g., forms being misplaced or damaged by water, etc.).

Cost effectiveness. With CAPI, a lot of the variable costs associated with traditional PAPI are virtually eliminated. The hiring of data entry staff and supervisors is no longer necessary or can be limited, since the data is already in digital form. Moreover, data cleaning at the end of project is greatly reduced because of the checks programmed into the tablets at the point of entering interview responses. In addition, some CAPI platforms are provided free of cost.

Additional data types. CAPI allows an interviewer to record location according to global positioning system (GPS) technology as well as take photographs using the camera function built into most modern mobile devices. Other CAPI platforms also allow for voice recordings when necessary.

Preloading of data. For longitudinal or follow-up surveys, data can be preloaded into the CAPI system to make matching easier and more accurate than PAPI.

Timeliness. CAPI data can be accessed, checked, and/or analyzed almost instantly by exporting it from the cloud server. Under PAPI, paper forms need to be transported back to a central location and entered into a computer before any data can be seen by survey administrators.

1.3 Hardware Requirements

Conducting a CAPI survey requires certain hardware (Table 1.1). While this can be a significant initial investment, all of the equipment can be used for many future projects, proving to be most cost-effective for longitudinal surveys. When deciding how much to spend, the expense of each piece of hardware has to be weighed against factors such as the financial resources available, how often the hardware will be used for survey work, and conditions in the field, among others. It is also worth considering the purchase of spare devices to cover for unforeseen events such as hardware failure, breakage, theft, and/or loss during fieldwork.

1.3.1 Tablets

Table 1.1: Tablet Specifications for CSEntry, June 2019

Specifics	Requirements
Type of Operating System	Android 4.0 or higher
RAM	Minimum of 1.5 gigabytes
Internal memory	At least 1 gigabyte of free memory to install the software
Wi-Fi	Required to be used for set-up, upgrades, and syncing
Connectivity	3G/4G
Screen Size	Depends on the requirement of the survey

RAM = random access memory

Source: United States Census Bureau. Census and Survey Processing System. <https://www.census.gov/data/software/cspro.html>.

Surveys requiring mobile internet would require tablets with subscriber identity module or subscriber identification module (SIM) card slots. Not all tablets contain a slot for a SIM card: those without are often called media tablets.

It is also advisable to purchase cases for tablets, allowing protection from excessive dust, dirt, moisture, falls, and other physical damage during fieldwork. The relatively low cost of cases is quickly recouped through extended tablet lifetime.

1.3.2 Power Solutions

To recharge tablets, secondary power sources may be necessary in the event of long interviewing days and/or remote locations. Mobile power options include car chargers (usually adapted to the vehicle's

cigarette lighter), portable lithium batteries or powerbanks, and solar chargers. Powerbanks are a recommended option given their portability and convenience.

1.3.3 Internet Connection

Each tablet can possess its own internet connection via a SIM card. SIM cards provide tablets with mobile internet access, which allows for data uploading to take place right after each survey is completed. However, mobile internet access can be limited and/or very costly in certain fieldwork areas. In these instances, another option might be to purchase a mobile router that shares the internet from one SIM connection to other devices using Wi-Fi. Alternatively, traditional Wi-Fi setups can be used to upload the survey data in the evenings, after fieldwork, if the interviewers are staying in a sizeable town or city. An option for when mobile internet is unavailable, usually in remote areas, would be to utilize Bluetooth to sync all of the interviewers' tablets to the supervisor's tablet. Once the datasets from the interviewers' tablets have been transferred to the supervisor's tablet, the supervisor then travels to a location where internet can be accessed to upload the survey data.

1.3.4 Personal Computers and Laptops

Personal computers (PCs) should be utilized by staff working in the head office, i.e., the staff members who will design and set up the entry system as well as those who will work on quality control during fieldwork. The minimal configuration for PCs to run CSPro is shown in Table 1.2.

Table 1.2: Hardware Specifications for CSPro

Specifics	Requirements
Operating System	Windows 7, 8, or 10
CPU	Pentium Processor
RAM	At least 512 megabytes
Internal Storage	At least 100 megabytes free storage on the hard disk

CPU = central processing unit, PC = personal computer, RAM = random access memory.

Source: United States Census Bureau. Census and Survey Processing System. <https://www.census.gov/data/software/cspro.html>.

1.3.5 Server

The final requirement for a CAPI survey is somewhere to store the data. CSPro offers the following data storage options:

- Dropbox
- A file transfer protocol (FTP) server
- A cloud-based or physical server with CSWeb installed.

1.4 CAPI Project Workflow

Migrating to CAPI entails an increased level of complexity in terms of designing the questionnaire and managing changes to it before fieldwork.

CAPI is NOT a replacement for a questionnaire developed in Microsoft Word or Excel. Rather, it should be seen as having the same function as a data entry system in a traditional paper-based survey. The CAPI system is programmed to not only match the developed questionnaire, but also to serve as a vehicle to input interview-derived data.

It is still necessary to design the questionnaire in either Microsoft Word or Excel because such software provides the best means to test how a survey will read and function on paper, before migrating the survey design to CAPI. Moreover, it may be helpful to use paper copies of your questionnaire for training purposes, and to have printouts on hand as a backup in case hardware problems occur during fieldwork.

When designing a CAPI-based survey, it may be necessary to plan and lay out additional specifications or assumptions that are required to conduct the survey, but are not explicitly stated in the paper version of the questionnaire. These may include:

- assignment of a variable ID for upfront coordination with data processing and/or analysis teams;

- specification of question type (i.e., single select, multi select, etc.);
- additional text to help or guide the interviewer if required (including formatting);
- specification of question dependencies (where a question depends on a response from a previous question or a condition);
- an expected range of valid responses on which consistency checks and/or error message logic are to be based; and
- prefilling, where a response can be predetermined based on a prior response or an external source.

When dealing with a survey that needs to be conducted in more than one language, it is important to always develop the questionnaire in a master language, then translate this master version to the other languages of interest.

The workflow steps for creating a CAPI survey are as follows:

- design the questionnaire in Microsoft Word or Excel, including CAPI-specific assumptions and/or considerations, in a master language;
- translate the master questionnaire into any additional languages;
- build the data entry system to match the master questionnaire's content and features;
- test and finalize the system to ensure it is working as intended in the master questionnaire, i.e., it is capturing the intended variables;
- insert the additional language versions into the system; and
- install the survey on tablets, ready for pilot testing, followed by actual fieldwork.

It is possible that changes will be required during the process described above. These may include revised wording of questions, additional

instructions, new questions, and updated skip patterns, among other things. In the event of such changes, it is important to respect the workflow outlined here, first making sure that the questionnaire in the master language reflects the changes, followed

by updating for additional languages. By doing so, all variants of the questionnaire (master on paper, additional languages on paper, and different language versions programmed for CAPI) will correspond correctly with one another.

Chapter 2: Introduction to CSPro

CSPro, a public domain software package, was first released in 2000 and has become the leading software for data entry of paper forms used in large scale surveys and censuses. CSPro was developed by the United States Census Bureau (USCB) and is supported by the bureau and ICF Macro (the organization that implements the Demographic and Health Surveys). Funding for the development and maintenance of CSPro is primarily provided by the United States Agency for International Development (USCB 2019).

CSPro was created by combining the features of two previous Microsoft Disk Operating System (DOS)-based systems – the Integrated Microcomputer Processing System and the Integrated System for Survey Analysis (IMPS and ISSA), both of which were text-based and required substantial programming skills to operate. CSPro was the first system of its kind to provide a visual interface, and it could be operated without the need

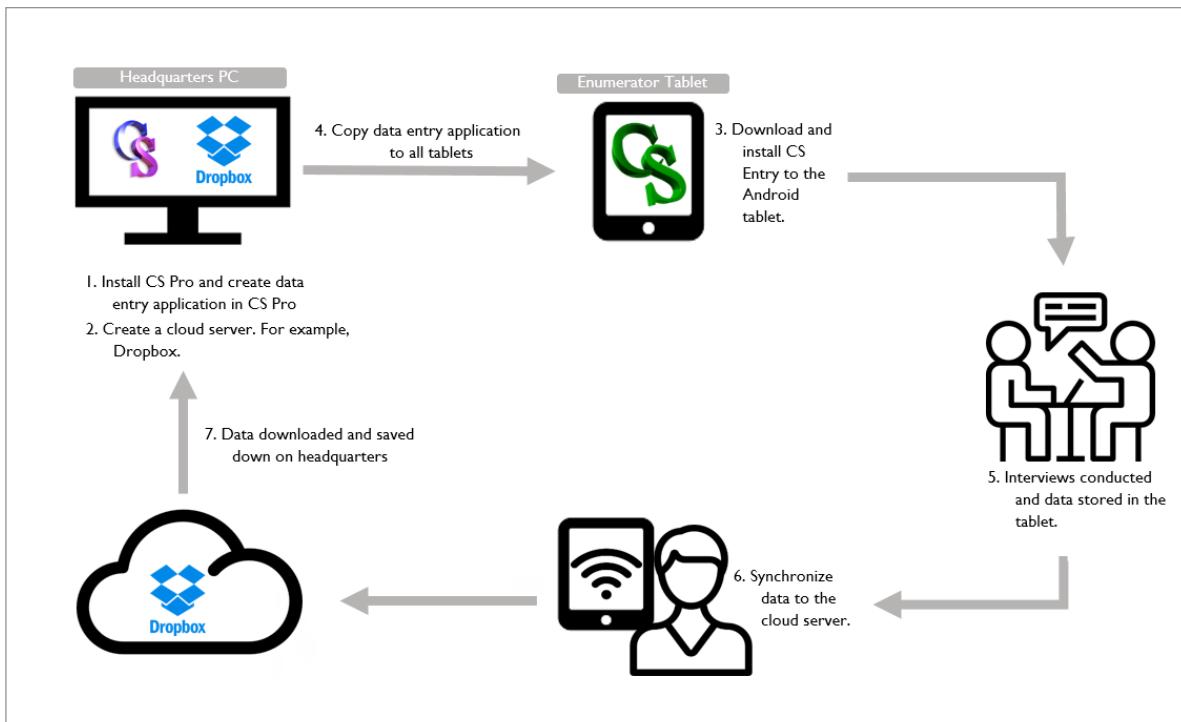
for high-level programming skills. As a result, it is now used by tens of thousands of organizations and individuals in more than 160 countries for entry, tabulating, and disseminating census and survey data (USCB 2019).

While CSPro contains many features, this handbook will focus on those necessary to build and set up a CAPI system for data collection, and to perform data quality control during CAPI fieldwork. The instructions include the use of CSEntry, the CSPro app that is installed on tablets and used specifically for inputting data. The handbook will cover CSEntry for Android, although it is important to note that a PC version also exists.

2.1 Data Flows in a CAPI Project

The data flows for a simple CAPI project using CSPro can be seen in Figure 2.1.

Figure 2.1: Data Flows in a Simple CAPI Project using CSPro



Step 1: Install CSPro onto a PC to create the Data Entry Application. This involves creating the data dictionary, forms, skips, and validations to make the questionnaire function correctly on a tablet.

Step 2: Identify and set up a place for the tablets to send (sync) collected data. This refers to a cloud server (e.g., Dropbox). The options for data storage are outlined in section 2.4.

Step 3: Install the CSEntry app on all Android tablets to be used for fieldwork.

Step 4: Deploy the CAPI questionnaire created on the PC to the fieldwork tablets. This can be accomplished through the cloud server or by using a universal serial bus (USB) cable to connect the PC to the tablets.

Step 5: Conduct interviews with respondents and record answers on the tablets.

Step 6: After interviews have been completed in the field, sync the data to the cloud server.

Step 7: Download the data from the cloud and save them locally on a PC or the server. The data can then be used for progress checks, quality control, or preliminary analysis.

2.2 Installing CSPro on a Personal Computer

To download CSPro for installation on a PC, go to the USCB website at <https://www.census.gov/data/software/cspro.Download.html> and click on the installation package for the latest version of CSPro (Figure 2.2).

You will be asked to complete a short form asking for some basic details. Once completed, press “Submit form” and the executable file will download (Figure 2.3).

Figure 2.2: Downloading CSPro for Installation on Personal Computer

The screenshot shows a website header with links for Overview, Documentation, Download (which is underlined in red), Related Sites, and Videos. Below this, a large blue header says "Download". Underneath, it says "Release Version Date" followed by "CSPro 7.1.3 Release Date: 2018-07-02". There are three download options: "CSPro 7.1.3 installation package" (with a blue outline), "CSEntry Android App", and "CSWeb [6.1 MB]". Below these are sections for "Video Based Instruction" (with a link to learn more about CSPro videos) and "CSPro E-Learning Videos" (with a link to watch official Census Bureau videos on YouTube).

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 2.3: Information Form to Download CSPro

The screenshot shows a form titled "CSPro Download Information". It asks the user to fill out the following fields: First Name, Last Name, Organization Name, Country (a dropdown menu), and E-mail. There is also a checkbox for "Please subscribe me to the CSPro mailing list so I can receive information about updates and other CSPro news." At the bottom are "Submit Form" and "Skip" buttons.

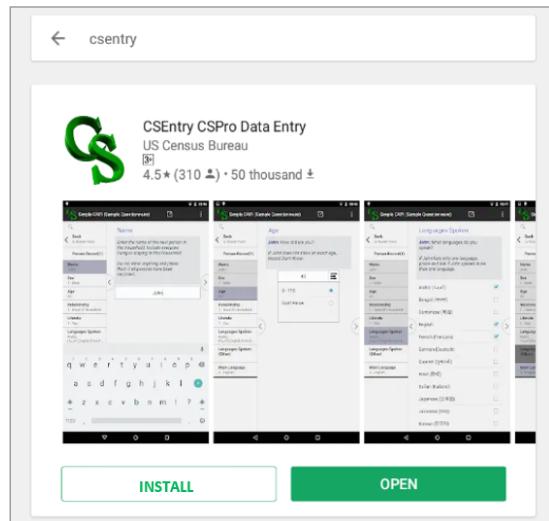
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Once the download is complete, run the executable file to install CSPro.

2.3 Installing CSEntry on a Tablet

The CSEntry app can be installed from the Google Play store by searching for “CSEntry”, then pressing “Install” (Figure 2.4).

Figure 2.4: Installing CSEntry from the Google Play Store



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

If you do not wish to set up and sign in to a Google Play store account on numerous tablets, a manual installation method is outlined in section 8.3.

2.4 Data Storage Options

CSPro offers three options for storage of survey data collected during a CAPI project.

2.4.1 Dropbox

Dropbox is a service that provides two gigabytes of cloud storage at no charge to the user. It is recommended for most CAPI projects because it is easy to set up, free to use, and integrates well with the CSPro system. To create a Dropbox account, go to <https://www.dropbox.com/>.

2.4.2 CSWeb

CSWeb is an app that can be set up on web-based servers, such as Amazon Web Services or Google Cloud, or on physical servers at the survey center or headquarters. Setting up CSWeb cannot be done independently and requires information technology (IT) staff to set up and test the server. This is typically recommended for large scale and complex CAPI projects or wherever there is a need for data to be stored within the country of survey. More information on CSWeb can be found on the CSPro Help website at <http://www.csprusers.org/help/CSWeb/>.

2.4.3 File Transfer Protocol Server

A local or cloud-based FTP server can be used with CSPro. An FTP server can be set up locally on a server at headquarters, or in the cloud with options such as FileZilla and Xflight.

2.5 CAPI Question Types

It is important to have an idea of the types of questions offered by CSPro and how interviewers can input data accordingly. When building a CAPI system, it is vital that the question type selected best captures the data of interest. CSPro offers the following question types for use in a CAPI survey:

2.5.1 Numeric Questions

Numeric questions allow the interviewer to input responses as numbers, using a numeric keypad, with both whole numbers and decimals allowed in CSEntry (Figure 2.5). Some examples of where this is applicable include questions relating to household identification (ID) number, respondent age, living costs, and spending habits.

Figure 2.5: Example of a Numeric Question in CSEntry

The screenshot shows a numeric input interface. At the top, there's a header for 'Simple CAPI (Sample Questionnaire)' with icons for search, back, and more. Below it, a section for 'Household ID' asks 'Enter the household number.' with a text input field. On the left, a sidebar lists 'Roster Form', 'Household ID' (selected), 'Interview Date', and 'Person Record'. At the bottom is a numeric keypad with a 'Next' button.

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

2.5.2 Text Questions

Text questions allow the interviewer to input responses in the form of alphanumeric characters and symbols in any language (Figure 2.6). This type of question is useful for recording the names of household members and “other”, “specify”, or open-ended questions.

2.5.3 Single Select Questions

Single select questions allow the interviewer to select one of at least two options (e.g., male or female) as shown in Figure 2.7. This type of question is useful for categorical questions where only one answer is possible, such as choice of gender or marital status.

2.5.4 Multi Select Questions

Multi select questions allow the interviewer to select one or more answers from a list of options (Figure 2.8). Such questions are useful where there can be more than one possible answer, such as languages spoken or foods eaten in the past 24 hours.

Figure 2.6: Example of a Text Question in CSEntry

The screenshot shows a text input interface. At the top, there's a header for 'Simple CAPI (Sample Questionnaire)' with icons for search, back, and more. Below it, a section for 'Name' asks 'Enter the name of the next person in the household. Include everyone living or staying in this household.' with a text input field. On the left, a sidebar lists 'Back to Roster Form', 'Person Record(1)', 'Name' (selected), 'Sex', 'Age', 'Relationship', 'Literate', 'Languages Spoken', 'Languages Spoken (Other)', and 'Main Language'. A keyboard is at the bottom.

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 2.7: Example of a Single Select Question in CSEntry

The screenshot shows a single select question interface. At the top, there's a header for 'Simple CAPI (Sample Questionnaire)' with icons for search, back, and more. Below it, a section for 'Sex' asks 'Jane: What is your sex?' with a text input field. On the left, a sidebar lists 'Back to Roster Form', 'Jane', 'Name Jane', 'Sex' (selected), 'Age', and 'Relationship'. Below the question, there are two radio buttons for 'Male' and 'Female'.

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

2.5.5 Date Questions

Date questions allow the interviewer to input a correct date using the calendar to scroll the days, months, and years (Figure 2.9). This type of question is useful for inputting dates, such as date of interview or dates of birth.

Figure 2.8: Example of a Multi Select question in CSEntry

Languages Spoken

Jane: What languages do you speak?
If Jane lists only one language, probe and ask if Jane speaks more than one language.

Arabic (العربية)	<input type="checkbox"/>
Bengali (বাংলা)	<input checked="" type="checkbox"/>
Cantonese (粵語)	<input type="checkbox"/>
English	<input checked="" type="checkbox"/>
French (Français)	<input type="checkbox"/>
German (Deutsch)	<input checked="" type="checkbox"/>
Gujarati (ગુજરાતી)	<input type="checkbox"/>
Hindi (हिन्दी)	<input type="checkbox"/>
Italian (Italiano)	<input type="checkbox"/>

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 2.9: Example of a Date Question in CSEntry

Interview Date

18	Oct	2017
19	Nov	2018
20	Dec	2019

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

2.6 Creating a CAPI Questionnaire in CSPro

The steps needed to set up a CAPI questionnaire using CSPro are as follows:

- Start a new project
- Set up the data dictionary
- Create forms and question text
- Add skips and validations
- Apply any additional languages
- Set up the questionnaire for data collection
- Establish quality control during fieldwork

These steps are expanded upon in the following chapters of this handbook.

Chapter 3: Starting a New Project

Start a new project by first clicking on the CSPro icon on the desktop (Figure 3.1).

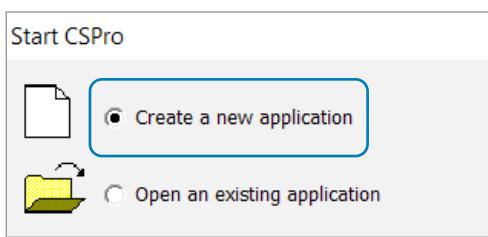
Figure 3.1: CSPro Desktop Icon



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

After clicking the CSPro icon, a prompt asks whether you would like to “Create a new application” or “Open an existing application” (Figure 3.2). For a new project, click on “Create a new application”.

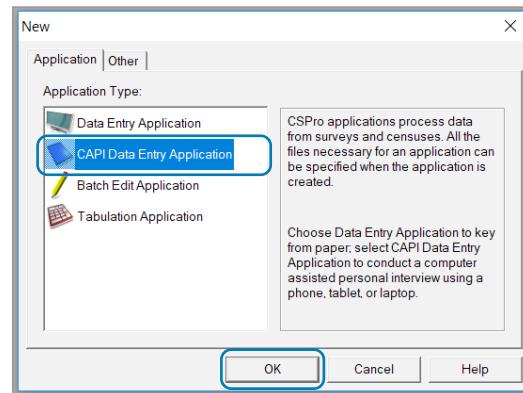
Figure 3.2: Creating a New Project in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Next, you will see another prompt with options for the “Application Type” as shown in Figure 3.3. To start a CAPI project, select the “CAPI Data Entry Application” option.

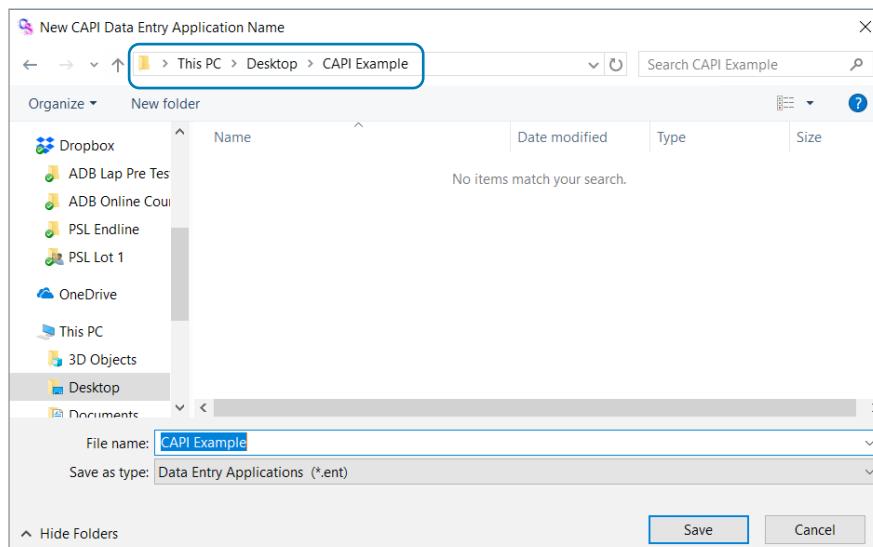
Figure 3.3: Creating a CAPI Survey in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

You will then be asked to give your CAPI survey a name, before saving it in the folder of your choice (Figure 3.4). The CSPro system consists of many files, so it is suggested that you create a new folder in which to store your CAPI survey.

Figure 3.4: Selecting the Directory and Naming the CAPI Survey



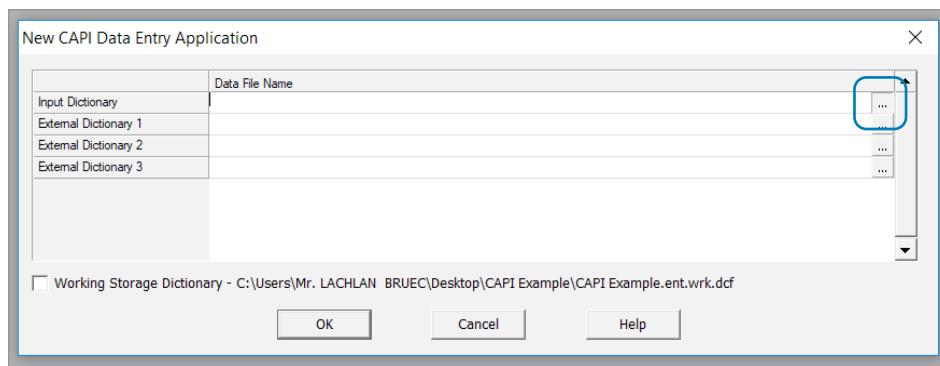
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The next step is to select a data dictionary for the survey project. The data dictionary is where the variables will be defined in order to store the data from the questionnaire. At this point, we will need to create a blank data dictionary to populate. Press the small grey “...” button to the right of “Input Dictionary” and create a new data dictionary in the same folder where you intend to store your survey (Figure 3.5).

Enter the name of the data dictionary (Figure 3.6). You can use the same name as the survey itself, since these two will have different file extensions.

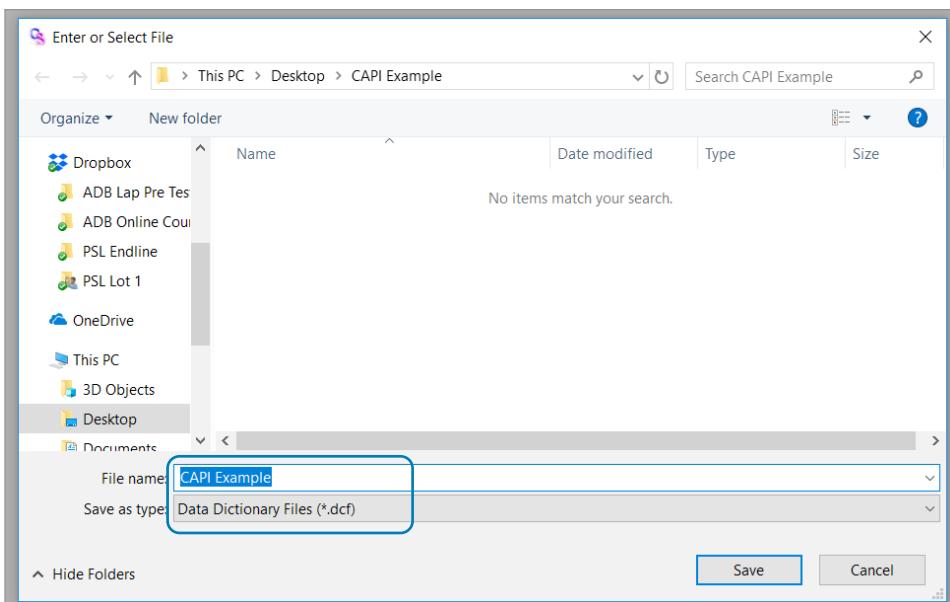
Now you have a new questionnaire and a blank data dictionary to start working on.

Figure 3.5: Creating a Blank Data Dictionary



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 3.6: Naming the New Data Dictionary File



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Chapter 4: Setting Up the Data Dictionary

The purpose of the data dictionary is to set up all the variables to be used in the CAPI project. Generally, a variable stores the answer from one question. The data dictionary primarily informs CSPro of the variable names, the data type of the variables (whether alpha, numeric, or alphanumeric), and how wide the variables need to be or how many characters are needed to store the information.

The data dictionary is what you first see after starting a new CAPI project. On the left of the screen is the tree view (Figure 4.1).

4.1 Identification Items

Every data dictionary needs to have “ID items”, so that each questionnaire to be answered can be uniquely identified. In simple paper-based data entry, the ID number could be a sequential number stamped on each form (1,2,3,4, so on). However, for a CAPI project, the ID is more complicated because questionnaires are being completed in different locations at the same time, so a single incrementing ID will not work. For CAPI, it is suggested that the

ID items are a combination of regional identifiers, e.g., the primary sampling unit (province, district, and village codes) plus the household ID. This way, the identifiers will be unique regardless of when the questionnaire responses are collected.

CSPro will provide an identification variable automatically, as shown in Figure 4.2.

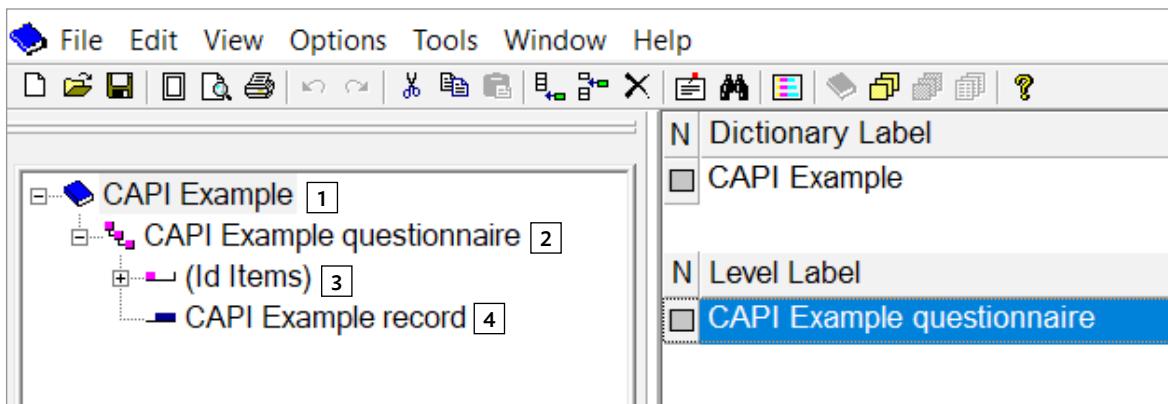
The ID item can then be modified by editing the fields on the right of the screen. These fields include the following:

Item Label. This label serves as the data output name. It can be optionally visible on the CAPI tablet screen in CSEntry.

Item Name. This serves as the name of the variable in the output data. It is also the name of the item to be referenced within the questionnaire when writing logic.

Start. This indicates the positioning of where the item starts in the data map. This field is automatically populated by CSPro.

Figure 4.1: Tree View in the Data Dictionary of CSPro



1 = dictionary (some projects and features require more than one), 2 = levels within dictionary (some project types require multiple levels),
3 = the identification items for the dictionary, 4 = records (different sections of the dictionary that will contain variables or items).

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.2: Identification Item in Tree and Item View

N	Item Label (record type)	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Ch	Zero Fill
	CAPI Example identification	CAPI_EXAMPLE_ID	2	1	Num	Item	1	0	No	No

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Length. This refers to the number of characters that will be reserved for an item. For example, a number ranging from 0 to 999 would require a length of three characters. An item for the name of respondent might have length of 50, allowing up to 50 alpha characters to be entered.

Data Type. This identifies whether the item will hold numbers (“Num”) or alphanumeric characters (“Alpha”).

Item. This denotes whether an item is a main item or a sub item. An item can generally be used for most variables in a data dictionary for a CAPI project. A sub item can be used to allow items to be broken up into smaller elements, e.g., day, month, and year of birth.

Occ. This refers to occurrences or the number of times a question will be asked.

Dec. This refers to the number of decimal places assigned to a numeric variable. Keep in mind that the decimal point itself will take a character space in the length column.

Dec Char. This specifies whether the item should be stored in the data file with an explicit decimal character.

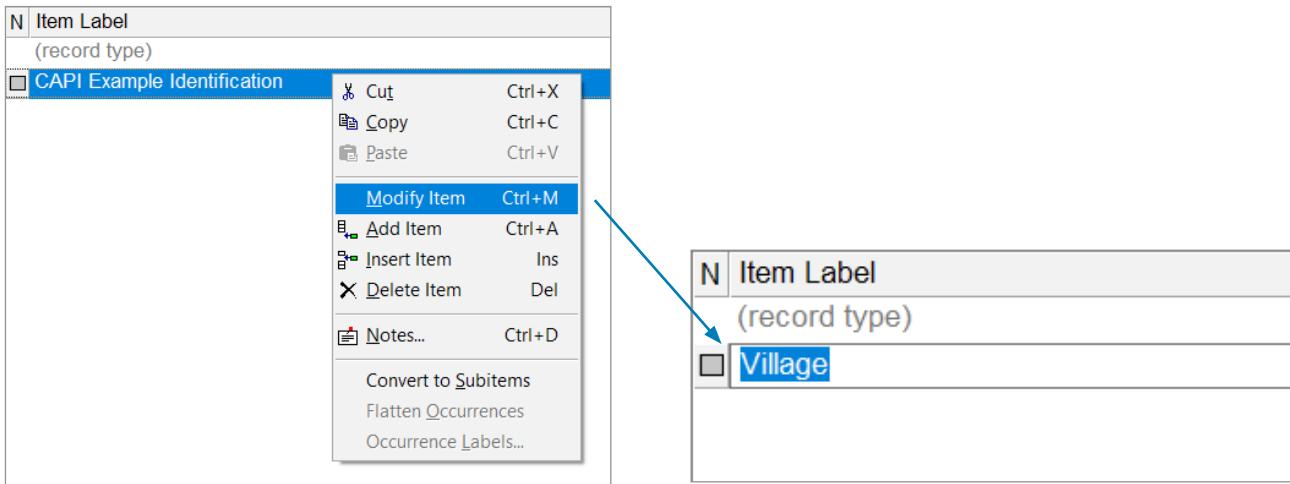
Zero Fill. This specifies whether the item will be stored with leading zeros, e.g., as “1” or “01”.

The ID items can be renamed and edited to match the first identifier, then any additional identifiers can be added. ID items and regular items can be modified, or new items added, by right-clicking on the item (Figure 4.3).

These identifiers need not be unique on their own, but their combined data code must be unique across all questionnaires to be answered (Figure 4.4).

Note that, in the example shown in Figure 4.4, there is a variable for “Village”. This variable is a numeric type because it holds a code for each village in the survey. The length has been set as “3” on the assumption that there will be fewer than 1,000 villages in the survey, each with its own code ranging from “1” to “999”. If there were more than 999 villages, but fewer than 10,000, then a length of

Figure 4.3: Modifying an Existing Identification Item



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.4: Multiple Identification Items

N	Item Label (record type)	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Ch	Zero F
	Village	VILLAGE	2	3	Num	Item	1	0	No	No
	Household ID	HOUSEHOLD_ID	5	2	Num	Item	1	0	No	No

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

“4” should be used. If your survey is being designed for a country or region with a vast number of villages, a higher length value would have to be set. For example, a survey covering 304,506 villages would require an item length of “6”. A second identifying variable, “HOUSEHOLD_ID”, has been added in Figure 4.4. When the two variables are combined, the data code for the village and household identifiers should always be unique.

4.2 Levels

Some complex surveys will require multiple levels in the data dictionary. These can be used where there are several questionnaires that need to be linked together. For example, a reproductive survey

could have different household-level questions—a questionnaire for each woman in the household, and another questionnaire about each pregnancy event for each woman.

4.3 Records

Records are like different sections (or chapters) of the questionnaire. For example, a separate record can be created for household demographics, and another for the household member roster (Figure 4.5).

The attributes to be set for each record are:

Record Label. This is a descriptive text label that identifies a record.

Figure 4.5: Adding Records to the Questionnaire

The screenshot shows the CSPro Data Dictionary window. On the left, there is a tree view of the project structure under 'CAPI Example' and 'CAPI Example questionnaire'. Under 'Household Demographics', there are 'Id Items', 'Household Demographics', and 'Household Roster'. On the right, a table lists records with columns: Record Label, Record Name, Type Value, and RequireMax. Two rows are visible: 'Household Demographics' with Record Name 'SECTIONA', Type Value '1', and RequireMax 'Yes 1'; and 'Household Roster' with Record Name 'SECTIONB', Type Value '2', and RequireMax 'Yes 1'.

N	Record Label	Record Name	Type Value	RequireMax
	Household Demographics	SECTIONA	1	Yes 1
	Household Roster	SECTIONB	2	Yes 1

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Record Name. This is a name given to a record for use in the CSPro logic.

Type Value. This is a numeric value to set the order of records in the data file.

Required. This specifies whether or not the record is mandatory.

Max. This refers to the maximum number of times a type of record can repeat for rosters.

and education are collected about each household member.

In some cases, the number of times the roster needs to be repeated is known. This type of roster is referred to as a fixed roster. For example, a block of questions is to be asked about five specific types of crops identified in a survey. In other instances, the number of times the block of questions is asked will depend on different survey cases. Take the case of a household roster where some households have three members, while others have six. This type of roster is referred to as a nonfixed roster.

It is important to note that a new record is created whenever a new type of roster is required. Therefore, if a block of questions (such as demographic questions) about the household members is required, that should be in one record.

To set up rosters in the data dictionary of CSPro, the “Max” value needs to be set to the maximum number of times each question will be repeated in the roster (Figure 4.7).

4.4 Rosters

A roster is a block of questions that need to be asked multiple times about multiple different subjects. On paper-based questionnaires, rosters are often presented as a grid, as illustrated in Figure 4.6. The most common example is the household roster from which demographics such as age, gender,

Figure 4.6: Simple Representation of Rostered Questions

The screenshot shows a grid table with four rows and three columns. The columns are labeled 'Gender' and 'Age'. The rows contain data for four individuals: Mathew (M, 45), Jane (F, 46), Billy (M, 14), and Mary (F, 11).

	Gender	Age
Mathew	M	45
Jane	F	46
Billy	M	14
Mary	F	11

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.7: Setting the Maximum Number of Roster Repeats

The screenshot shows the CSPro Data Dictionary window. The table on the right has a row for 'Household Roster' with 'Type Value' '2' and 'RequireMax' 'Yes'. A dropdown menu is open next to 'RequireMax', showing 'Yes' and 'BC'.

N	Record Label	Record Name	Type Value	RequireMax
	Household Demographics	SECTIONA	1	Yes 1
	Household Roster	SECTIONB	2	Yes ▾ BC

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

For nonfixed rosters, it is important to set the “Max” value to allow for the highest number expected as well as some allowance for extraordinary cases. For example, if the highest number of household members is expected to be 15, the “Max” value could be set at 20 or 30 to provide a buffer in case a household is found with an unusually large number of members. It is generally better to allow more space in the “Max” field than is required because, if the value is too low, the interviewer will not be able to enter complete information.

If the roster is a fixed type, where the names of each item are known before the survey starts, the occurrence labels (roster rows) need to be set in the data dictionary. To do this, right-click on the desired record and select “Occurrence Labels” (Figure 4.8).

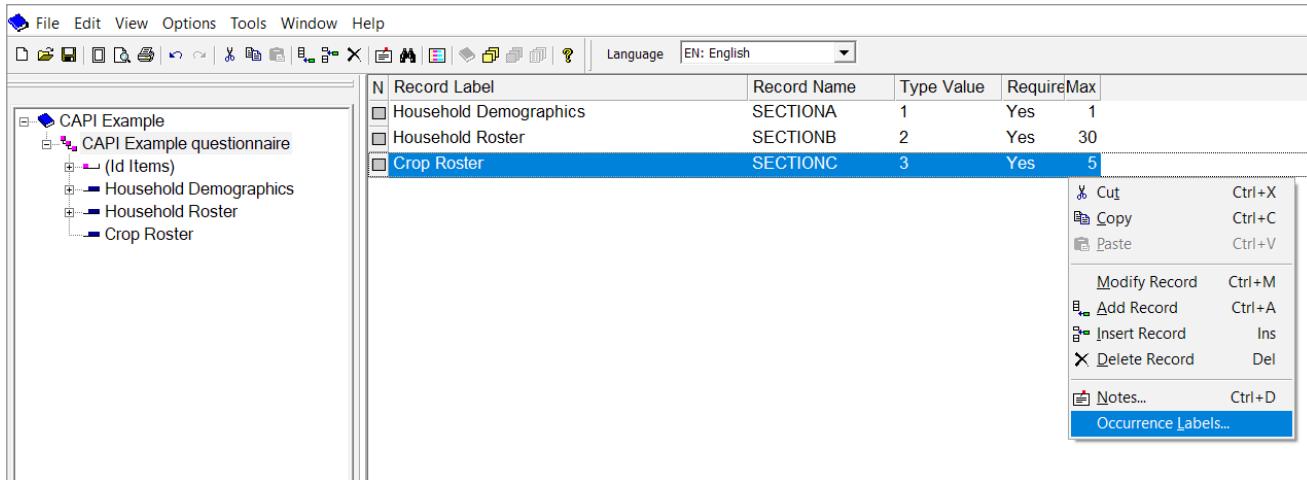
The names of each roster row can now be entered as the occurrence labels (Figure 4.9).

4.5 Items

An item is created for each record in the survey (regardless of how many times it is rostered) and any hidden variables required (Figure 4.10). By clicking on the record (e.g., “Household Roster”) in the tree on the left of screen, items (e.g., “Village” and “Household ID”) can then be added on the right of screen.

Once added, the items are displayed in the tree in a light-blue color. Subitems can be used to allow items to be broken up into smaller elements, e.g., day, month, and year of birthdate.

Figure 4.8: Setting Occurrence Labels for a Rostered Record

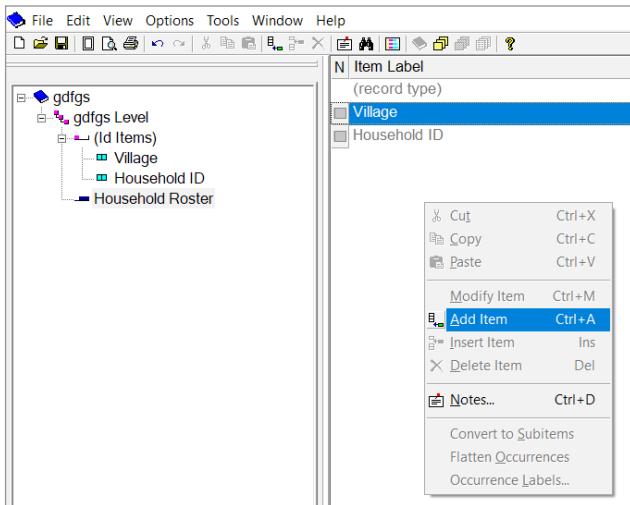


Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.9: Example of Completed Occurrence Labels

Occurrence Labels (Crop Roster)	
Occ	Label
1	Rice
2	Potato
3	Carrot
4	Cassava
5	Wheat

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.10: Adding an Item to a Record

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

4.5.1 Text Questions

In Figure 4.11, an item “B1 – Name” has been added to the household roster. It has been set as an “Alpha” data type with a length of “30”, meaning that text can be entered to a maximum length of 30 characters. It is recommended to devise a naming

convention for the variables in your data dictionary to correspond with the master version of the questionnaire. In Figure 4.11, since “B1” is the label given to a question asking household member name, it has been included in both the item label and the item name.

4.5.2 Numeric Questions

To set up a numeric question in the data dictionary, the data type must be set to “Num” and the length should be set to the maximum length expected. In Figure 4.12, for the item regarding age, the length is set to three digits. If the value is required to decimal places, the “Dec” column can be used to specify how many decimal places are required. Remember, however, that the length variable needs to be adjusted to accommodate the overall number of digits, including the decimal point.

4.5.3 Single Select Questions

A single select question is usually set as a numeric data type because it captures a numeric value corresponding to each answer option (Figure

Figure 4.11: Adding an Item for a Text Question

N	Item Label	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Chi	Zero	F
	(record type)			1	1	Alpha					
	Village	VILLAGE	2	3	Num	Item	1	0	No	No	
	Household ID	HOUSEHOLD_ID	5	2	Num	Item	1	0	No	No	
	B1 - Name	B1	7	30	Alpha	Item	1	0	No	No	

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.12: Adding an Item for a Numeric Question

N	Item Label	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Chi	Zero	F
	(record type)			1	1	Alpha					
	Village	VILLAGE	2	3	Num	Item	1	0	No	No	
	Household ID	HOUSEHOLD_ID	5	2	Num	Item	1	0	No	No	
	B1 - Name	B1	7	30	Alpha	Item	1	0	No	No	
	B2 - How old is [NAME]	B2	37	3	Num	Item	1	0	No	No	

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

4.13). If there are no more than 9 answer options, the length variable can be set to “1”. If, however, more than 9 options are present, the length should be set to “2”. It is recommended to set the length to “2” in any case, as it allows for more answer options to be added later and enables the use of a two-digit code for options such as “Don’t know” or “Others”.

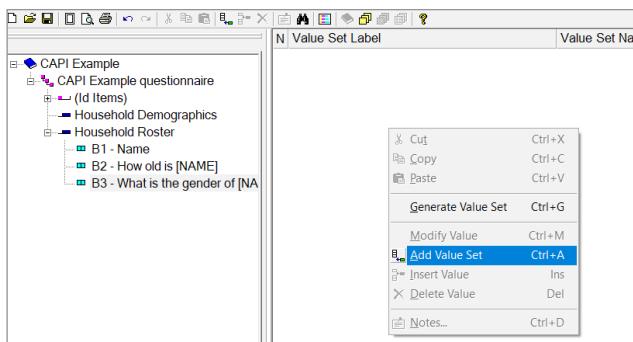
Figure 4.13: Adding an Item for a Single Select Question

N	Item Label	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Ch	Zero F
	(record type)		1	1	Alpha					
	Village	VILLAGE	2	3	Num	Item	1	0	No	No
	Household ID	HOUSEHOLD_ID	5	2	Num	Item	1	0	No	No
	B1 - Name	B1	7	30	Alpha	Item	1	0	No	No
	B2 - How old is [NAME]	B2	37	3	Num	Item	1	0	No	No
	B3 - What is the gender of [NAME]	B3	40	2	Num	Item	1	0	No	No

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

A “value set” refers to all possible answer options for a specific question. To add a value set for a single select question, click on the item in the tree on the left of screen. Then, in the value set section on the right of screen, right-click and select “Add Value Set” (Figure 4.14).

Figure 4.14: Adding a Value Set for a Single Select Question



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Value sets have the following properties:

Value Set Label. This is a label for the value set that will appear in tabulation applications.

Value Set Name. This serves as reference to the value set to be used in CSPro logic.

Value Label. This is the answer option text that will appear on the CAPI screen and as the label in output data.

From & To. These specify the numeric range of values for each value set, for example allowing respondent age of 0 to 120. Leaving the “To” column blank will leave only one value for each (used for single and multi select questions).

Special. This can be used to set missing, default, or not applicable values, e.g., when a respondent refuses to answer or does not know the answer to a question.

Image. This can be used to add an image to each answer option.

The Value Set Label can match the text that will be displayed on the screen to interviewers (e.g., “Male” or “Female”, coupled with their corresponding unique values “1” and “2” as seen in Figure 4.15).

Value sets can be modified by right-clicking on an existing set and selecting “Modify Value Set” (Figure 4.16).

Figure 4.15: Example of a Value Set for a Single Select Question

The screenshot shows the CSPro software interface. On the left, there is a tree view of the questionnaire structure under 'CAPI Example' and 'CAPI Example questionnaire'. On the right, a table titled 'Value Set' is displayed. It has columns for 'Value Set Label', 'Value Set Nar', 'Value Label', 'From', 'To', 'Special', and 'Image'. One row is selected, labeled 'B3 - VS1'. The 'Value Label' column contains 'Male' and 'Female'. The 'From' column is '1' and the 'To' column is '2'. There is also a 'Special' column.

N	Value Set Label	Value Set Nar	Value Label	From	To	Special	Image
	B3 - VS1		Male	1	2		

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.16: Modifying an Existing Value Set

The screenshot shows the CSPro software interface with a context menu open over a value set row. The menu options include: Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Format Value Pair Labels, Replace Value Labels (Paste from Clipboard), Make Primary Value Set, Generate Value Set (Ctrl+G), Modify Value Set (selected), Add Value Set (Ctrl+A), Insert Value Set (Ins), Delete Value Set (Del), and Notes... (Ctrl+D).

N	Value Set Label	Value Set Nar	Value Label	From	To	Special	Image
	B3 - VS1		Male	1	2		

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

4.5.4 Multi Select Questions

A multi select question should be set up as an “Alpha” data type variable. The length variable should be equal to the number of answer options in the question. For example, a question on the languages spoken by a given household member is accompanied by a list of 28 possible language options. As shown in Figure 4.17, the item for such a question would be set to “Alpha” data type and given a length of “28”.

The value set is then created in the same way as for a single select question, except that letters are used for the values instead of numbers (Figure 4.18). The reason for this is that all answer options selected will be stored in the variable as, for example, “ACF” if the options A, C, and F are chosen. If numbers were to be used, the output data for the example discussed would be “136”, making it impossible to tell if options 1,3, and 6 were selected, or if the selections were in fact options 13 and 6.

Figure 4.17: Adding an Item for a Multi Select Question

The screenshot shows the CSPro software interface. On the left, there is a tree view of the questionnaire structure under 'CAPI Example' and 'CAPI Example questionnaire'. On the right, a table titled 'Item' is displayed. It has columns for 'Item Label', 'Item Name', 'Start', 'Len', 'Data Typ', 'Item Type', 'Occ', 'Dec', 'Dec Ch', 'Zero', and 'Fi'. One row is selected, labeled 'B4'. The 'Item Label' column contains 'B4 - What language(s) does [NAME] speak?'. The 'Item Name' column is 'B4'. The 'Start' column is '42', 'Len' is '28', 'Data Typ' is 'Alpha', 'Item Type' is 'Item', 'Occ' is '1', 'Dec' is '0', 'Dec Ch' is 'No', and 'Zero' is 'No'.

N	Item Label	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Ch	Zero	Fi
	B4 - What language(s) does [NAME] speak?	B4	42	28	Alpha	Item	1	0	No	No	

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Any alphanumeric characters can be used in value sets. What's more, because these characters are case sensitive, up to 75 possible combinations can be achieved using a-z, A-Z, and 13 symbols—!@#\$%^&*()[]{}.

4.5.5 Date Questions

A date question can be set up in the data dictionary as an item of numeric data type with

a length of “8” (Figure 4.19). This way there are sufficient digits to store the values for year, month, and day as “YYYYMMDD”. A common example of a date question is one that seeks a respondent's date of birth.

After setting up the data dictionary, click on save.

Figure 4.18: Example of a Value Set for a Multi Select Question

N	Value Set Label	Value Set Nar	Value Label	From	To
	B4 – What language(s) does [NAME] speak?	B4_VS1			
			Arabic (العربية)	A	
			Bengali (বাংলা)	B	
			Cantonese (粵語)	C	
			English	D	
			French (Français)	E	
			German (Deutsch)	F	
			Gujarati (ગુજરાતી)	G	
			Hindi (हिन्दी)	H	
			Italian (Italiano)	I	
			Japanese (日本語)	J	
			Javanese (ꦗꦮ)	K	
			Korean (한국어)	L	
			Malay (Bahasa Melayu)	M	
			Mandarin (官話)	N	
			Marathi (मराठी)	O	
			Persian (فارسی)	P	
			Portuguese (Português)	Q	
			Punjabi (ਪੰਜਾਬੀ)	R	
			Russian (Русский)	S	
			Spanish (Español)	T	
			Tamil (தமிழ்)	U	
			Telugu (తెలుగు)	V	
			Thai (ไทย)	W	
			Turkish (Türkçe)	X	
			Urdu (اردو)	Y	
			Vietnamese (Tiếng Việt)	Z	
			Wu (吳語)	0	
			Other Language	1	

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 4.19: Adding an Item for a Date Question

	Item Label	Item Name	Start	Len	Data Typ	Item Type	Occ	Dec	Dec Chi	Zero	F
CAPI Example	(record type)		1	1	Alpha						
CAPI Example questionnaire	Village	VILLAGE	2	3	Num	Item	1	0	No	No	
(Id Items)	Household ID	HOUSEHOLD_ID	5	2	Num	Item	1	0	No	No	
Household Demographics	B1 - Name	B1	7	30	Alpha	Item	1	0	No	No	
Household Roster	B2 - How old is [NAME]	B2	37	3	Num	Item	1	0	No	No	
B1 - Name	B3 - What is the gender of [NAME]	B3	40	2	Num	Item	1	0	No	No	
B2 - How old is [NAME]	B4 - What language(s) does [NAME] speak?	B4	42	28	Alpha	Item	1	0	No	No	
B3 - What is the gender of [NAME]	B5 - What is [NAME]'s date of Birth?	B5	70	8	Num	Item	1	0	No	No	
B4 - What language(s) does [NAME] speak?											
B5 - What is [NAME]'s date of Birth?											

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Chapter 5: Creating Forms and Question Text

5.1 Creating Forms for CAPI

In CSPro, forms are the visual interface used by the interviewer to enter values for the items that have been set up in the data dictionary. Creating forms for a CAPI project is much easier than for a desktop data entry project because all the formatting is taken care of automatically for CAPI (when creating forms for desktop data entry, the formatting needs to be done manually to ensure the questionnaire is user friendly and has the same layout as the paper form).

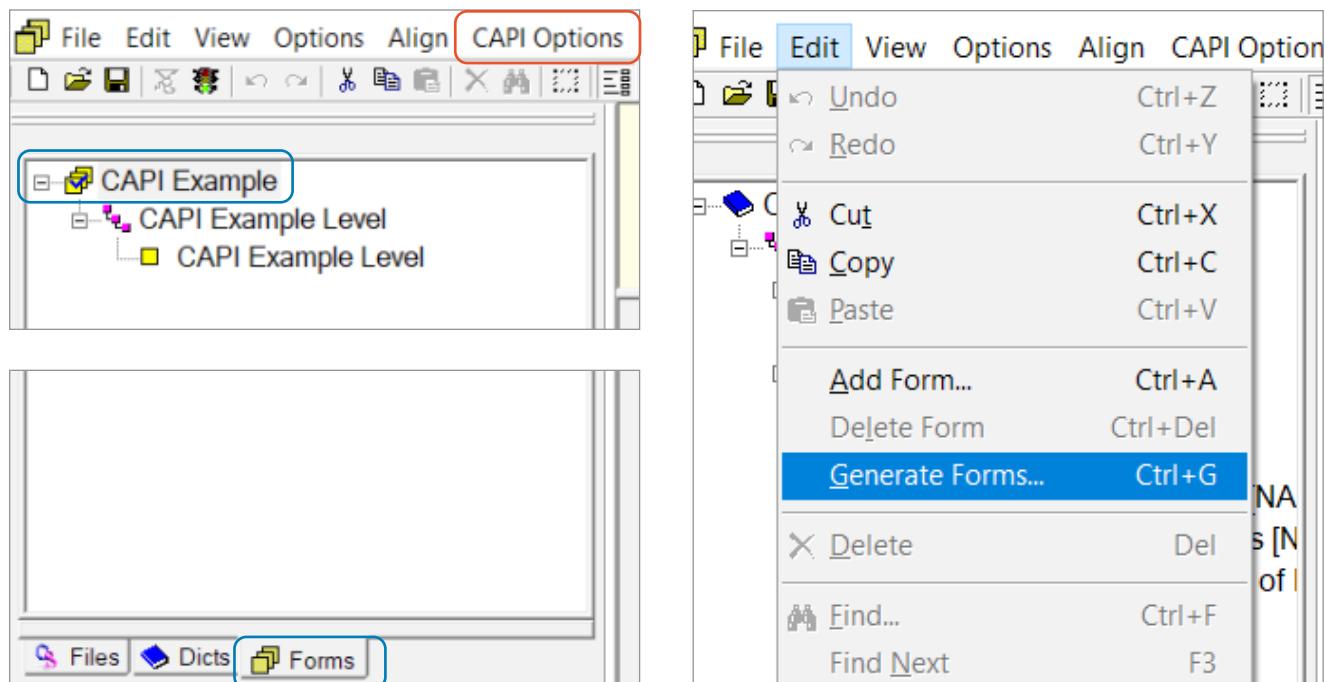
To create the forms for CAPI, first click on the “Forms” tab at the bottom of screen (Figure 5.1).

Next, double-click on the default form group, then go to the “Edit” menu and select “Generate Forms”.

Once you have clicked “Yes”, the forms options dialogue box will appear (Figure 5.2). For a CAPI project, you may accept all the default settings and press “OK”.

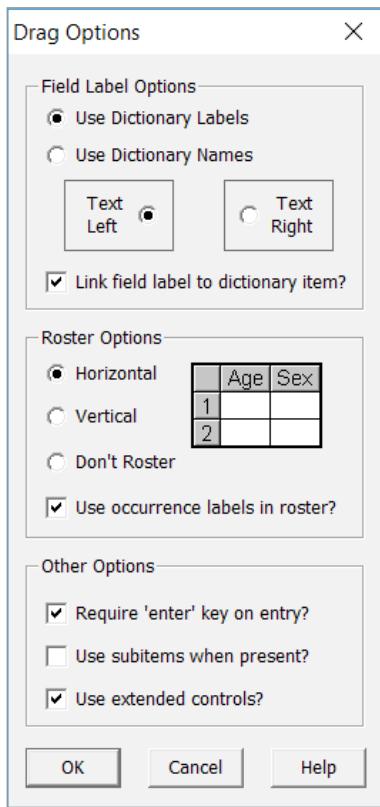
The forms are then created automatically. You can navigate between your files, the data dictionary, and the forms by using the tabs at the bottom left of screen (Figure 5.3).

Figure 5.1: Generating Forms for a CAPI Project



CAPI = computer-assisted personal interviewing

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 5.2: Options Dialogue Box when Creating Forms

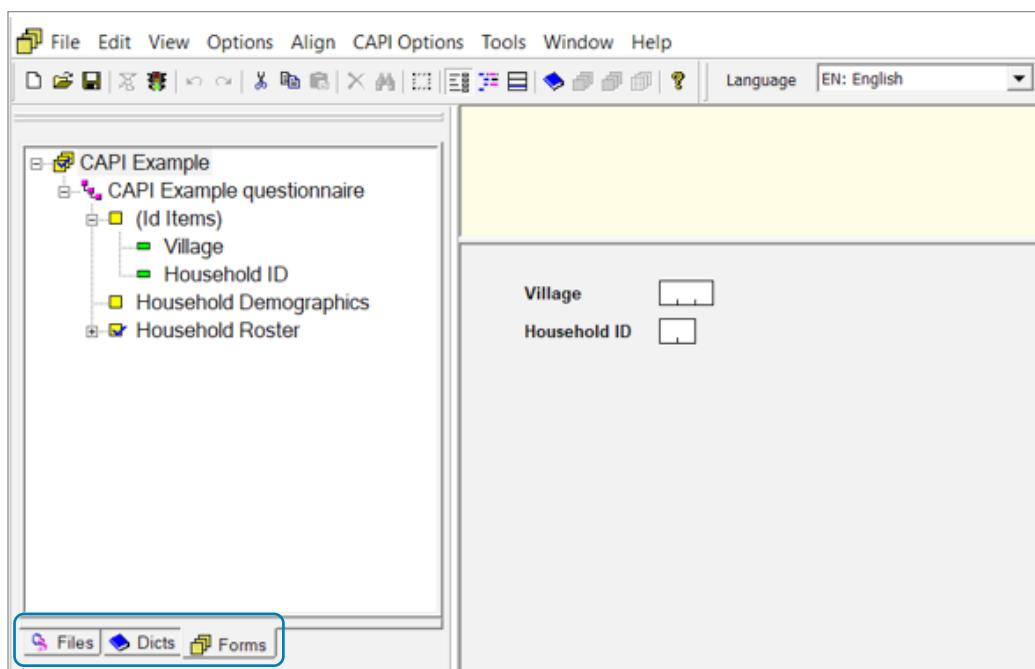
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

On the left of screen, you will notice a tree view for all the forms created (Figure 5.4). One form is created for the ID items, following by a form for each record. On the right of screen, you can see the form with all the items from that record displayed. It is possible to format the forms on the right for use on PC, but for CAPI this is not required.

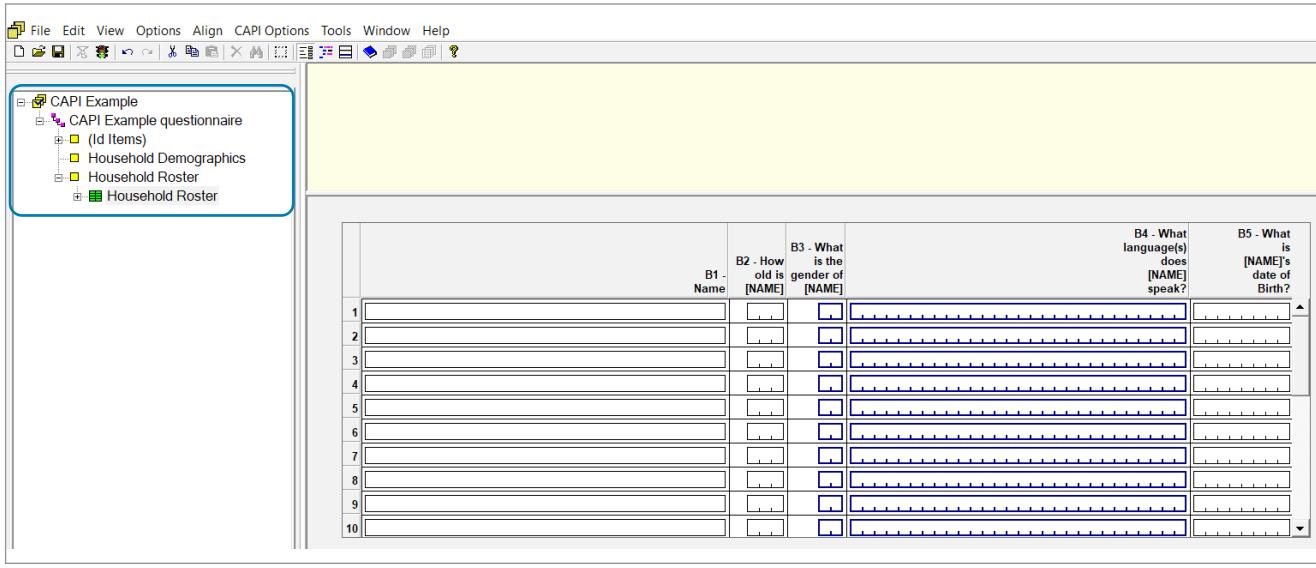
You now have a basic scaffold of your questionnaire ready. The remaining steps are to add functionality to it.

5.2 Blocks

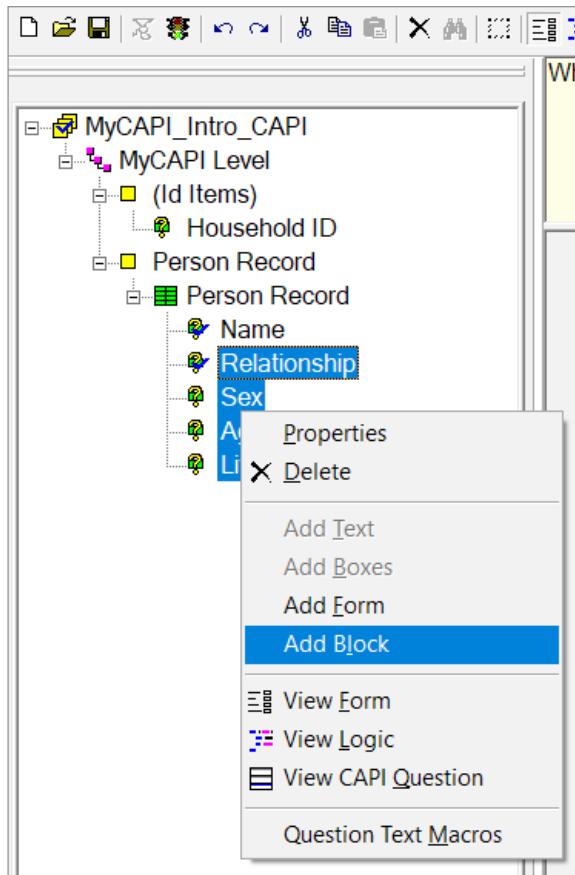
“Blocks” can be used after forms have been created to show multiple questions on one screen of the CAPI questionnaire. To create a block, go to the form tree on the left of screen, hold down the “Ctrl” key on your keyboard, then click on the necessary questions to be displayed on one screen (Figure 5.5). Once chosen, right-click and select “Add Block”.

Figure 5.3: Navigating between Files, Dictionaries, and Forms

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

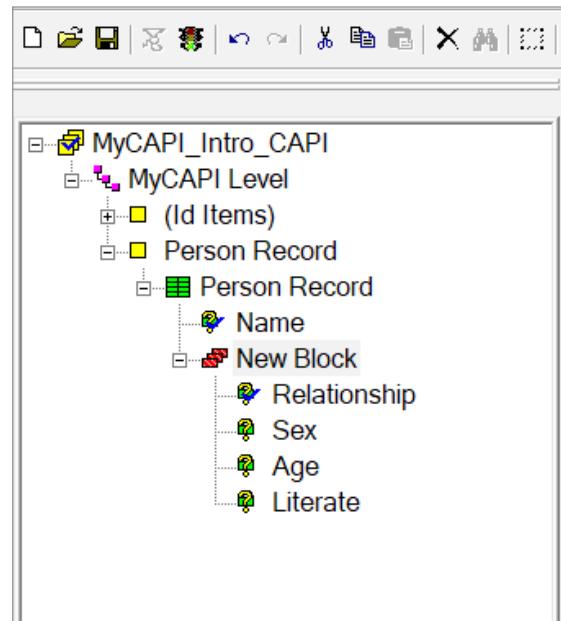
Figure 5.4: Tree View of Forms and Listing of Items

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 5.5: Adding Blocks to Forms

Source: Screen shot generated by Asian Development Bank consultant. 2019. Newcastle, United Kingdom.

After assigning a name to the block, the questions will be displayed under a red block in the form tree (Figure 5.6). In the CSEntry app on the tablets, these questions will display together on one screen.

Figure 5.6: Example of Questions Placed into a Block

Source: Screen shot generated by Asian Development Bank consultant. 2019. Newcastle, United Kingdom.

5.3 Adding New Questions

In some cases, a new question will be added to the questionnaire after the forms have been created. In this case, the question needs to be first added to the data dictionary. If data collection has already started, the best practice is to add the new item at the end of the record, so that it does not move the location of existing items (Figure 5.7).

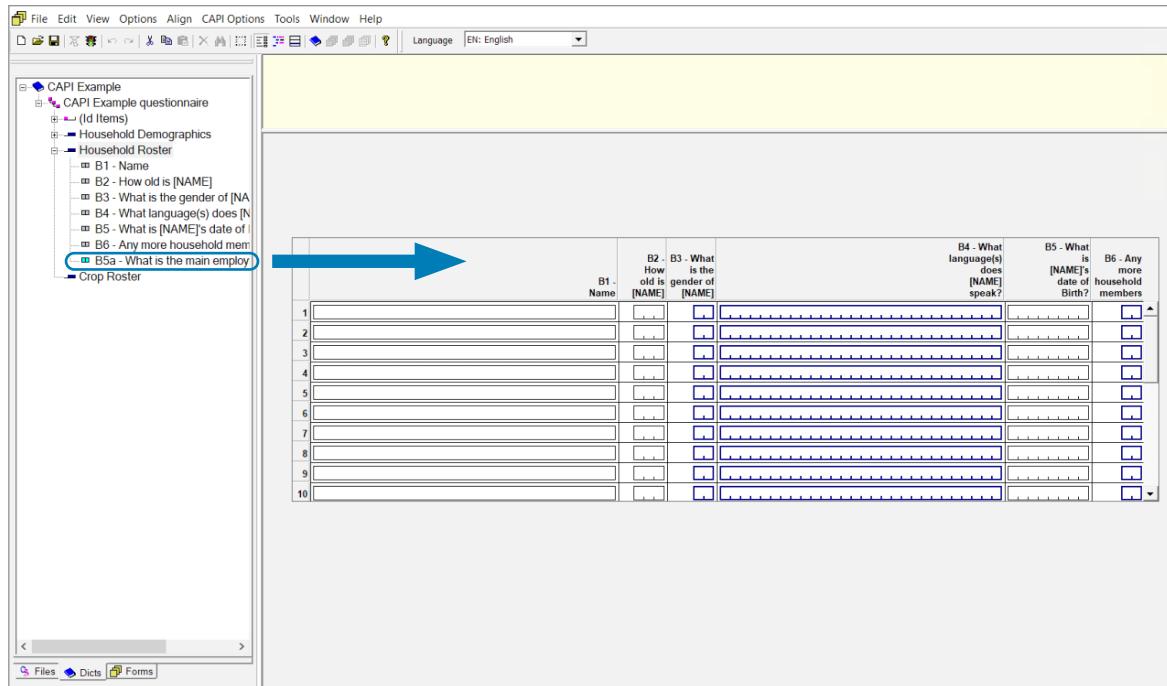
Next, navigate to the relevant form where the question was added, then move the left pane back to the data dictionary using the tab at the bottom of screen, so the screen looks like the example in Figure 5.8. You will notice in the data dictionary tree that the recently added variable is colored blue, indicating it is not present in any form.

Figure 5.7: Adding a New Question to the Data Dictionary

N	Item Label (record type)	Item Name
<input checked="" type="checkbox"/>	Village	VILLAGE
<input checked="" type="checkbox"/>	Household ID	HOUSEHOLD_ID
<input checked="" type="checkbox"/>	B1 - Name	B1
<input checked="" type="checkbox"/>	B2 - How old is [NAME]	B2
<input checked="" type="checkbox"/>	B3 - What is the gender of [NAME]	B3
<input checked="" type="checkbox"/>	B4 - What language(s) does [NAME] speak?	B4
<input checked="" type="checkbox"/>	B5 - What is [NAME]'s date of Birth?	B5
<input checked="" type="checkbox"/>	B6 - Any more household members	B6
<input checked="" type="checkbox"/>	B5a - What is the main employment activity of [NAME]	B5A

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

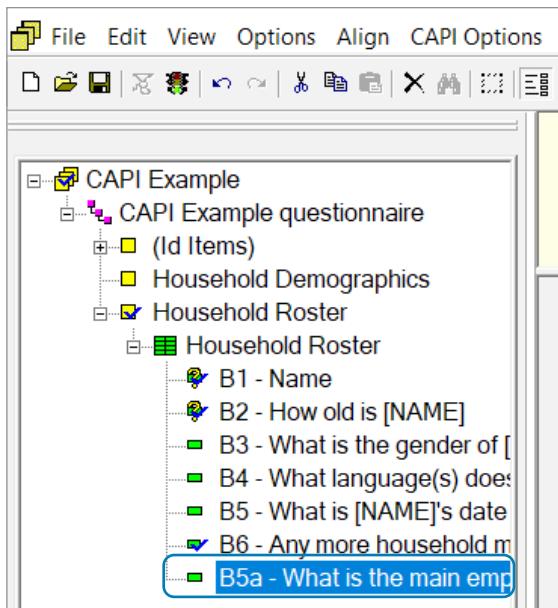
Figure 5.8: Dragging a Newly Added Question from the Dictionary to the Forms



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Drag the new item from the data dictionary across to the form, then navigate back to the form tree to correct the order of the new item. By default, the newly added item will be the last to display on that form (Figure 5.9).

Figure 5.9: Default Order of Newly Added Dictionary Item on Form



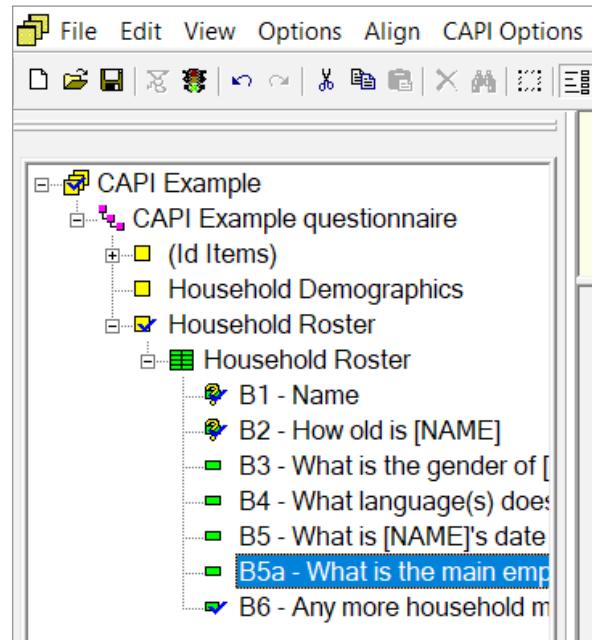
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Drag the new item up to its correct order in the questionnaire flow (Figure 5.10).

5.4 Creating Question Text and Interviewer Instructions

CAPI question text and interviewer instructions will appear on the tablet screen for the interviewers to read. There are two distinct fonts available in CSPro and, ideally, one font should be used for the question text (to be read aloud) and another font for the interviewer instructions (usually to be read in silence).

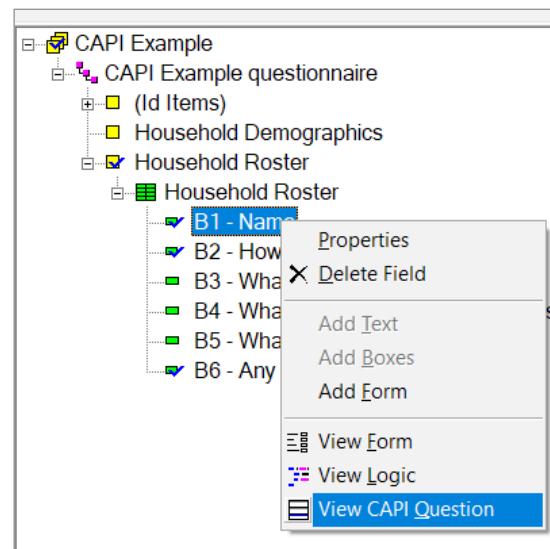
Figure 5.10: Corrected Order of Newly Added Dictionary Item on Form



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

The question text can be created or edited by right-clicking on a question in the form tree and selecting “View CAPI Question” (Figure 5.11). It is good a practice to copy and paste the question text from the master questionnaire to avoid typing mistakes.

Figure 5.11: Viewing and Editing Question Text



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

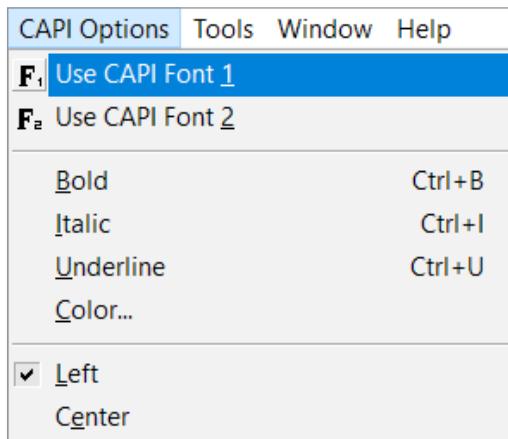
Using the “CAPI Options” menu, the text can be set as either of the two fonts available (Figure 5.12).

The text for all questions and any required instructions can then be created in the same way.

5.4.1 Setting Help Text

Where an interviewer might require additional guidance on a given question, help text can be added by pressing on the green “?” icon in the text formatting panel (Figure 5.13). This allows the

Figure 5.12: Setting the Font of the Question Text



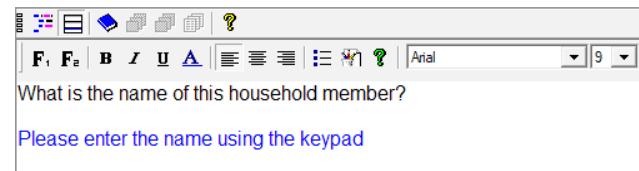
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

interviewers' manual to be incorporated into the CAPI questionnaire. The help text does not need to be explicitly displayed for each question, but can be accessed by the interviewer if needed.

5.4.2 Dynamic Text Display in Roster Questions

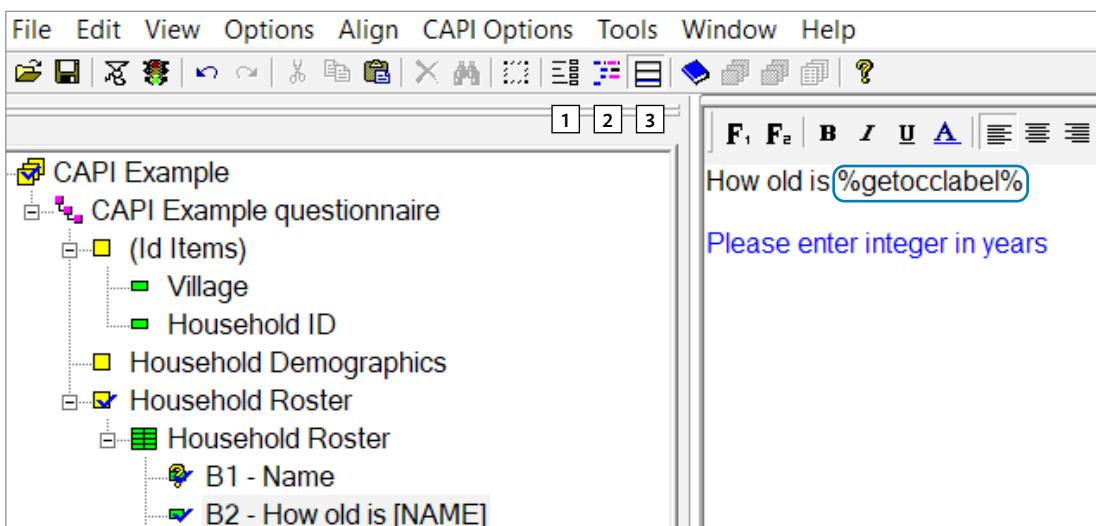
For questions that are rostered, the occurrence label can be dynamically displayed in the question text, meaning the text will change to display the relevant subject each time the question is asked. To insert the occurrence label of the roster, the code “%getocclabel%” can be inserted in the relevant position (Figure 5.14). For fixed rosters, the occurrence labels will be set in the data dictionary. For nonfixed rosters, the labels will be setup dynamically in the system logic (section 6.7).

Figure 5.13: Adding Help Text to a Question



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 5.14: Dynamically Displaying the Occurrence Label



1 = form view, 2 = logic, 3 = question text.

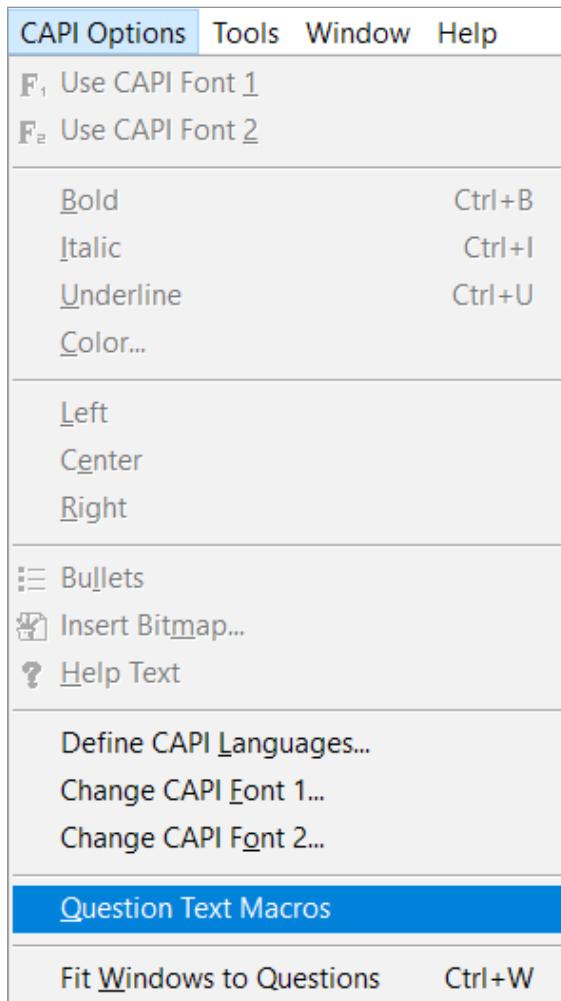
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

In the forms section, you can use the icons to move between form view, logic, and question text as shown in Figure 5.14.

5.4.3 Question Text Macros

To make the setting of question text easier, and to avoid errors, macros can be utilized. These can be accessed by clicking on “CAPI Options” and selecting “Question Text Macros” (Figure 5.15)

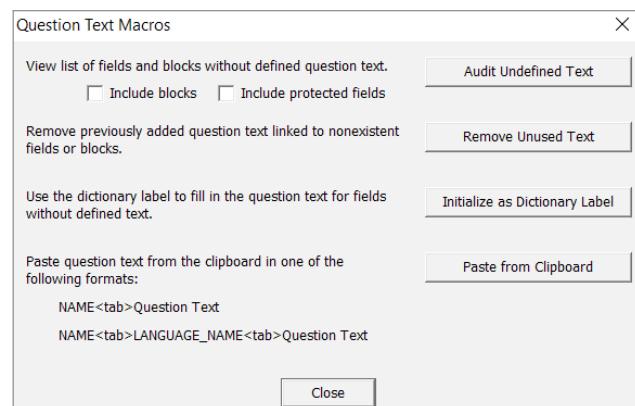
Figure 5.15: Accessing Question Text Macros



Source: Screen shot generated by Asian Development Bank consultant. 2019.
Newcastle, United Kingdom.

Macros allow for copying and pasting from a text file, and can be used to apply checks where required to find any missing text in the questionnaire (Figure 5.16).

Figure 5.16: Dialogue Box for Question Text Macros



Source: Screen shot generated by Asian Development Bank consultant. 2019.
Newcastle, United Kingdom.

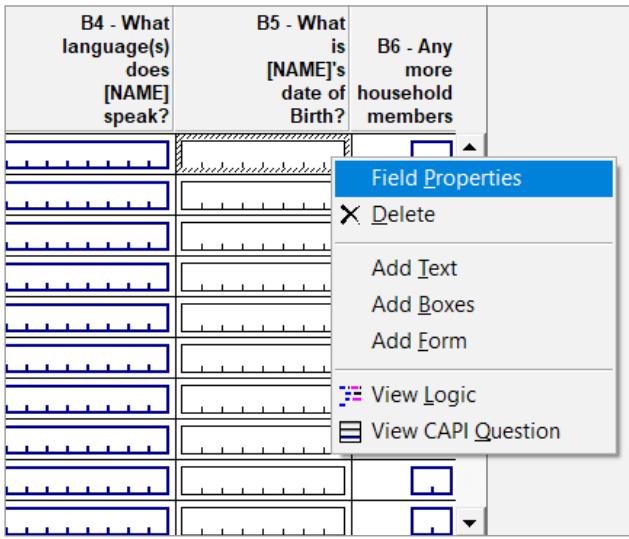
5.5 Displaying the Calendar for Date Questions

The appearance of most question types will be set automatically, depending on how the data dictionary is defined. Date questions are exceptions because you are required to change some settings to have the response panel displayed as a calendar and not a numeric input box.

To do this, find the relevant date question in the form, right-click on its input box, and select “Field Properties” (Figure 5.17).

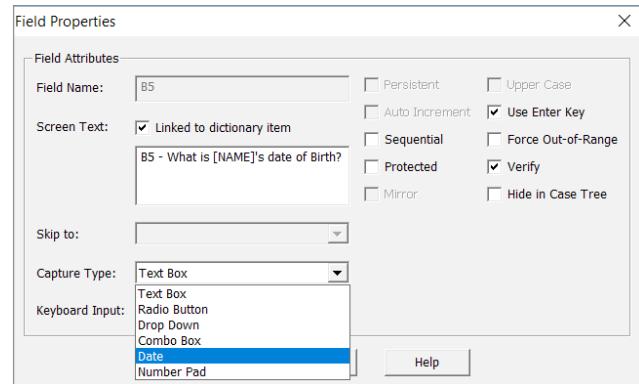
In the “Field Properties” window, change the capture type to “Date” and press “OK” (Figure 5.18).

Figure 5.17: Applying Field Properties to a Date Question



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 5.18: Modifying the Capture Type for a Date Question



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 6: Skips and Validations

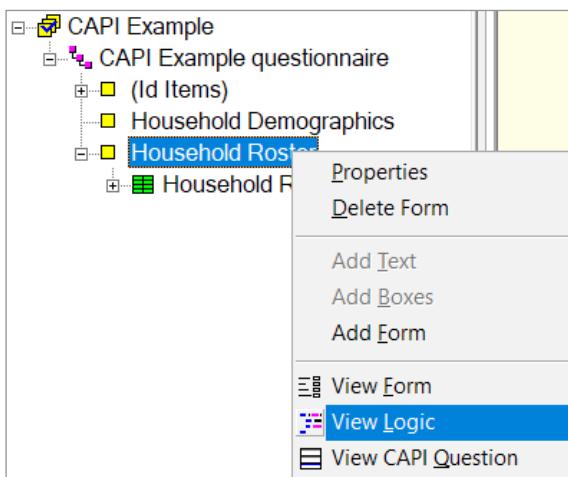
6.1 An Overview of Skips

Similar to paper-based surveys, CAPI surveys can have “skips”. Sometimes referred to as “enabling conditions”, “skip patterns”, or “questionnaire logic”, skips are used to allow a question to remain unasked depending on the type of respondent or the answers to previous questions. On paper-based forms, a written “if... then” instruction is typically given to the interviewer on whether to ask the next question or to jump forward in the questionnaire.

CAPI can make this process easier by programming the skips into the system. This means that questions are only displayed if certain conditions are satisfied, and the interviewer only needs to ask the questions the system displays. One major advantage of this is that interviewers can focus on their interviewing techniques, asking only valid questions, and without having to think about skips. A CAPI system is also generally able to incorporate more sophisticated skips than those typically possible using traditional methods.

Skips in CSPro need to be written in the “Logic” sections of the form editor. To access the system logic, enter the form view, then click on the “View Logic” button (Figure 6.1).

Figure 6.1: Navigating to the System Logic in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

6.2 Logic Structure

6.2.1 Global vs Procedural

The first step in writing logic code is to determine the correct place in CSPro to enter it. There are different logic sections in which code can be entered and initiated at different times. First, there is “global” or “declaration” section. This is the section where the variables or functions to be used throughout the program are defined. The global logic section can be accessed by selecting the topmost form in the tree on the left of screen (Figure 6.2).

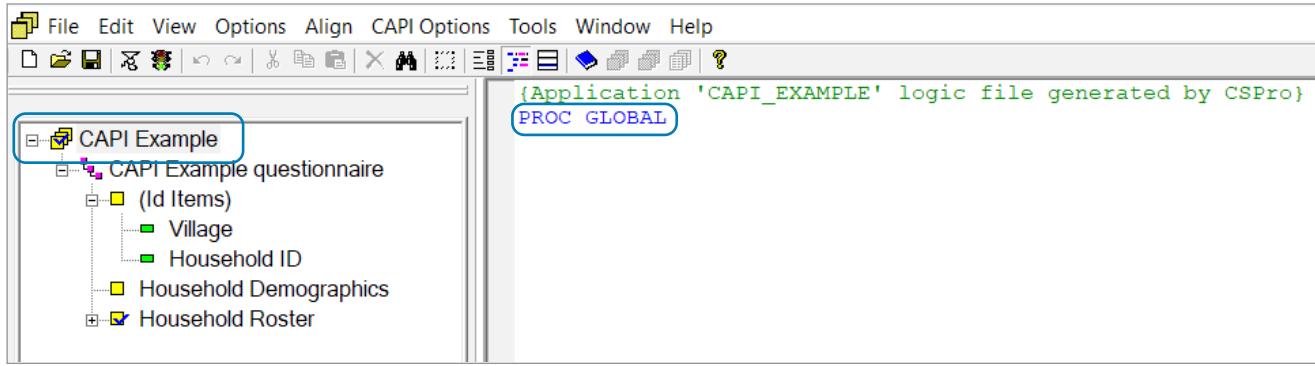
All other logic sections are called “local” or “procedural” sections. Code in this section will be executed as the entry reaches them. For simple skips, the code will usually be inserted into the procedural sections of a question, such as “B1” in Figure 6.3. In some instances, more complicated logic has to be inserted into the form or roster procedural sections. You can navigate around the different sections of logic by clicking on the tree on the left of screen.

6.2.2 Procedural Types

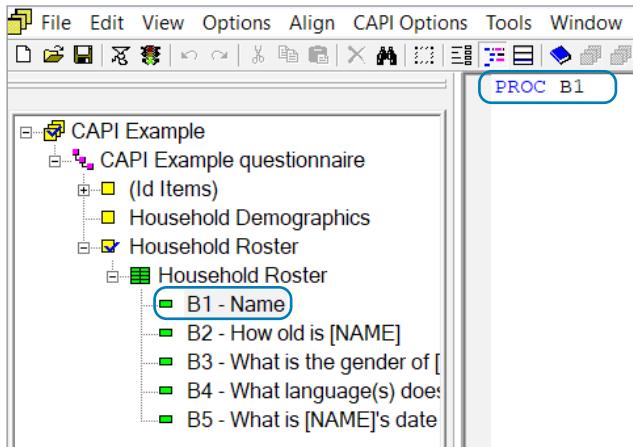
CSPro offers different types of procedural logic, which can determine when the logic code will be executed.

PostProc. The logic code in PostProc runs after the answer to the question has been entered and the interviewer has pressed the forward arrow to advance to the next section. This is the default in CSPro and, if no procedural type is specified, the logic code will be run as PostProc.

PreProc. By specifying PreProc, the logic code will run as the system moves forward and arrives at the relevant question, roster, or form.

Figure 6.2: Accessing the Global Logic Section

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.3: Example of a Procedural Logic Section

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

OnFocus. This is similar to PreProc, except that the logic code will be executed on arriving at the relevant question when moving either backward or forward in the questionnaire. If PreProc is also present, OnFocus will run after PreProc when moving forward in the questionnaire.

KillFocus. This is the opposite of OnFocus. The logic code will be executed as the interviewer leaves a question, travelling either forward or backward.

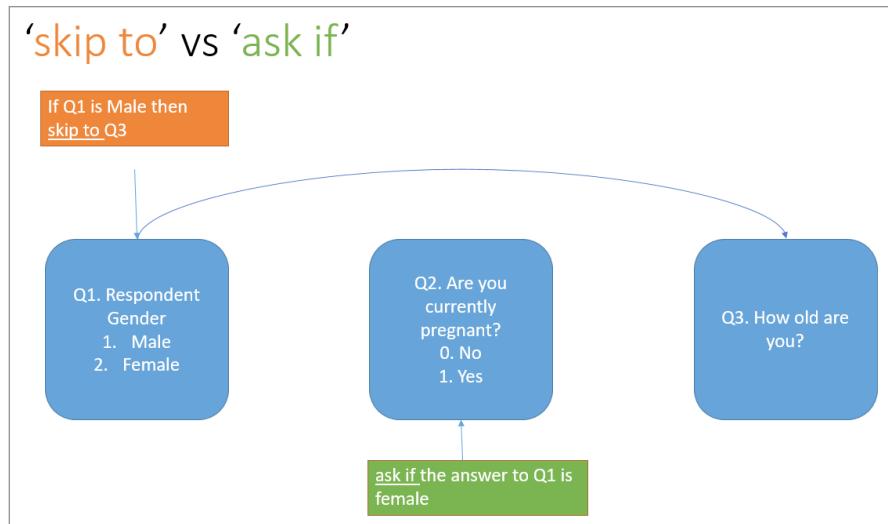
Onocchange. This runs when the occurrence of a roster changes.

6.3 Basic Syntax for Implementing Skips

The two basic logic commands used for implementing skips in CSPro are the “ask if” and “skip to” commands. Figure 6.4 demonstrates the difference between the two functions: “skip to” is more useful if the interviewer is required to move far ahead in the survey, whereas “ask if” allows a question to be shown depending on the response to the previous question or a condition further back in the questionnaire.

Next, we will discuss using a basic “if” statement in the logic code as shown in Figure 6.5.

An example of an “if” statement for a basic “skip to” command is shown in Figure 6.6. The logic code is written in the procedural section of the item named “Q1” (asking the respondent’s gender). Since no procedural type has been specified, the code will run as the default, PostProc, meaning it will be executed after Q1 has been completed. The condition used in the example includes “\$”, which references the current question. The full condition will be “\$=1”, since “1” is male in the value set for this question. The action to be taken in this instance is to “skip to Q3” (asking the respondent’s age).

Figure 6.4: Basic Logic Commands for Skips

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.5: Using an “If” Statement in Logic Code

```
Proc Q1
if [CONDITION] then
    [ACTION];
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 6.6: Example of Using an “If” Statement to Create a Skip

```
Proc Q1
if $=1 then
    skip to Q3;
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

The same result can be achieved by using an “ask if” command to create a skip (Figure 6.7). Here the relevant question for the logic code is “Q2” (asking if the respondent is pregnant). For this “ask if” command, the logic code needs to run before Q2 is answered. To do this, we use the PreProc procedural type. Figure 6.7 below shows the completed example to create a skip using ‘ask if’.

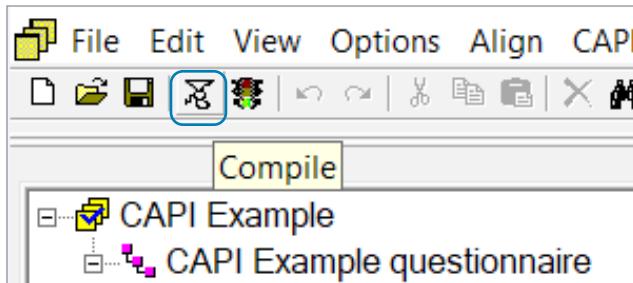
Figure 6.7: Example of Using an “Ask If” Command to Create a Skip

```
Proc Q2
PreProc
ask if Q1=2
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

After writing your logic code, always compile the code to check for errors. This can be done by clicking the “Compile” button on the icon bar (Figure 6.8).

Figure 6.8: Using the Compile Button to Check for Errors



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

If an error is found during compilation, a dialogue box will appear, informing you that the compile has failed (Figure 6.9). A brief description of the error is shown at the base of screen and a red

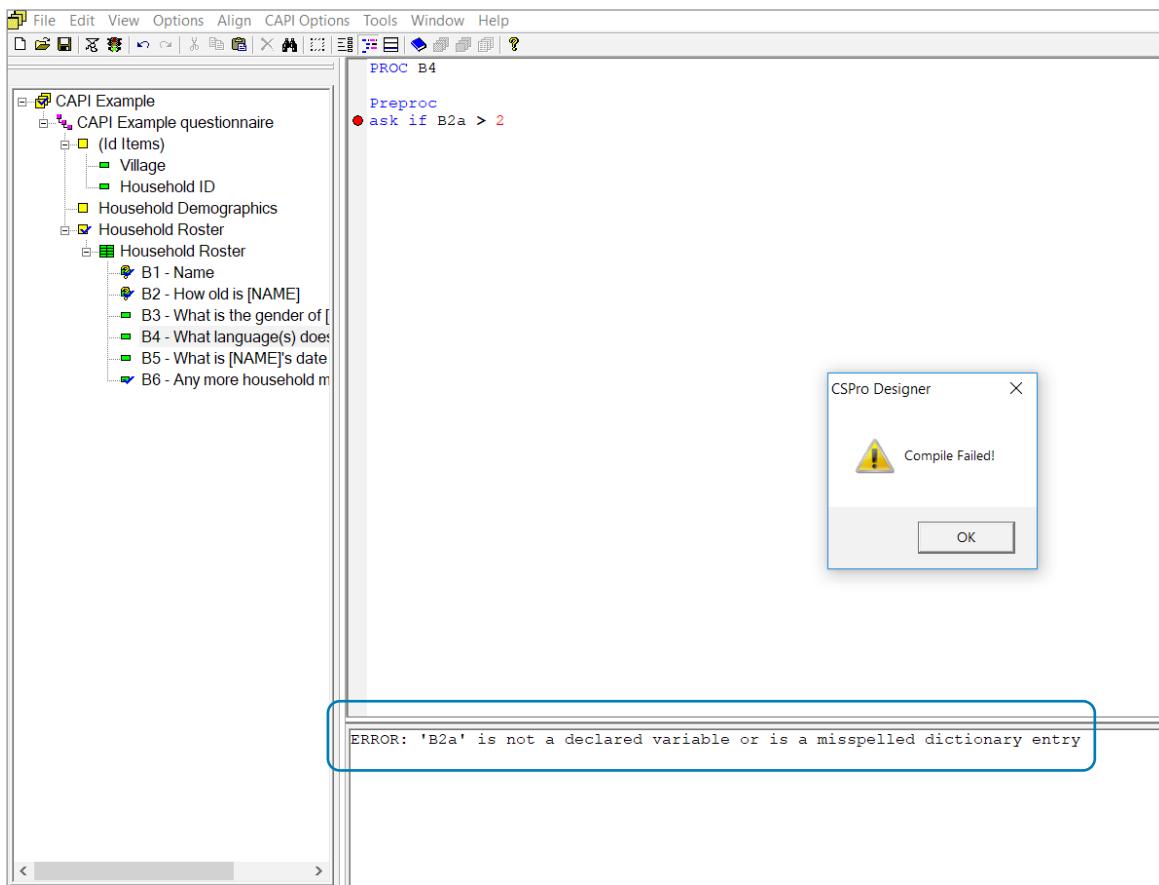
circle will appear next to the line of code with the error (a yellow circle appears to advise of a warning).

In the example in Figure 6.9, the wrong name for the variable was specified, i.e., “B2a” instead of “B2”. Once the error or errors have been corrected in the code, a message will be displayed at the base of screen to advise that the compilation has been successful (Figure 6.10).

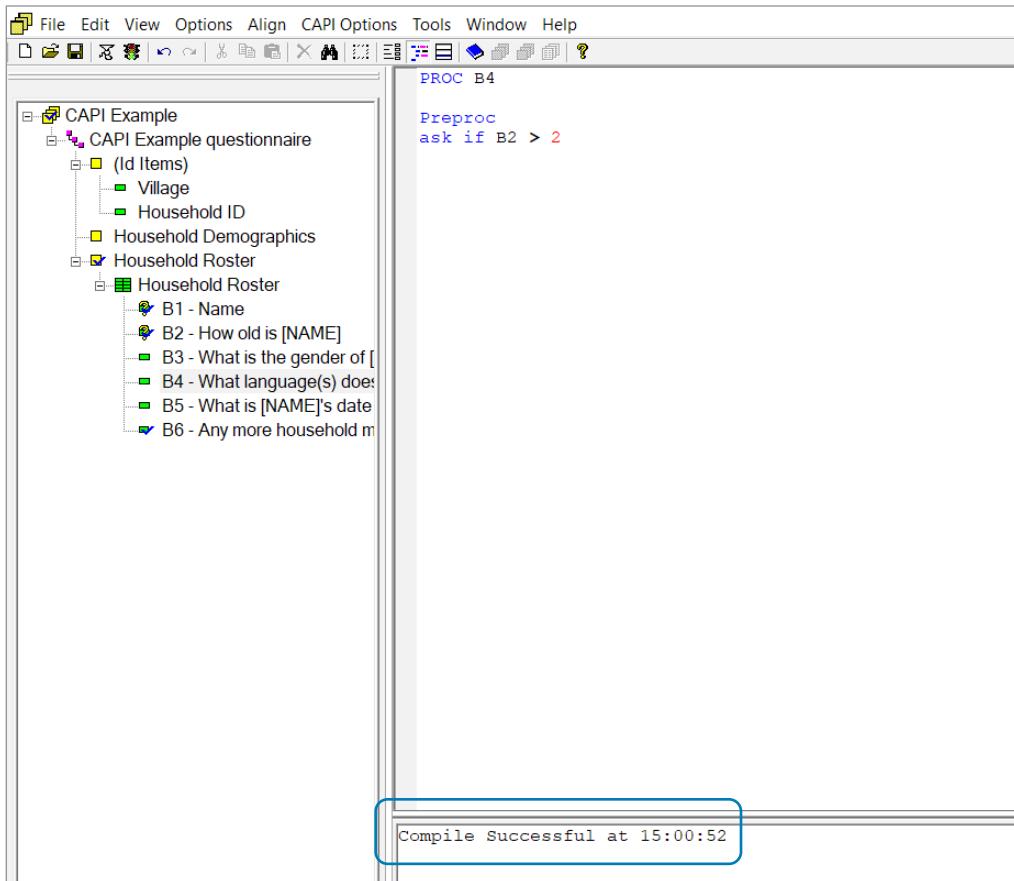
6.4 Other Useful Movement Commands

The skip command will move the user forward in the survey, leaving the questions unasked and any logic code unexecuted. Some other useful commands include:

Figure 6.9: Example of a Compilation Error in Logic Code



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.10: Example of Successfully Compiled Logic Code

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

“Move to”. This allows backward or forward movements in the questionnaire, and is useful if you have a userbar¹ to navigate to different sections.

“Advance to”. This allows forward movement in the questionnaire, but will execute any logic code in the questions it moves past. This is useful for questionnaires with hidden questions or values that are to be prepopulated by logic.

6.5 Relational and Logical Operators

When working with single select or numeric questions, relational operators can be used to create conditions (Table 6.1). For example, “AGE > 17” will include all respondents of ages 18 and above, while “Of GENDER = 1” will specify males (if the code “1” is designated for males in the value set).

Table 6.1: Relational Operators Used in CSPro

Operator	Meaning
=	Is equal to
<>	Is not equal to
>	Is greater than
>=	Is greater than or equal to
<	Is less than
<=	Is less than or equal to
in xy	In the range of x and y
\$	Current Item

¹ Setting up a userbar is covered in chapter 10 of this manual.

In some instances, the condition to be implemented is more complicated and depends on the answer to two or more questions. In these cases, logical operators can be used to combine conditions (Table 6.2). For example, for a question to be asked only to females who are older than 15, the code could be written as: “Ask if GENDER = 2 && AGE >= 15”.

Table 6.2: Logical Operators Used in CSPro

Operator	Meaning
&&	AND
	OR
!	NOT

Source: Table constructed by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

6.6 Multi Select Skips

Since multi select questions store data using alpha characters (letters) rather than numeric characters (numbers), relational operators cannot be applied to them. To demonstrate how we overcome this, let’s use the example below:

Q1. Which of the following does your household own?

- A. Bicycle
- B. Motorbike
- C. Car
- D. Truck
- E. None of these

Q2. How many cars does your household own?

The second question can only be answered if option C was selected as one of the options in the first question. If option “C” was selected, we expect the response to the first question to contain the letter “C” in its string, with potential response combinations being “ACD” (household owns at least one bicycle, car, and truck), “BC” (household owns at least one motorbike and car), or just “C” (household owns at least one car and none of the other options). To search through the “Q1” variable to look for “C”, we use the “pos” function in CSPro (Figure 6.11). The

query string will be the value you want to check for, while the source string will be the question you want to search in.

Figure 6.11: Creating Code to Search for Alpha Characters

```
pos(query string, source string);
```

Will return the position in the string where the query is and a 0 if it is not in the string

e.g pos(“C”, “ACDE”) will return value of 2

e.g. pos(“G”, “ACDE”) will return 0

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

In the example in Figure 6.12, The “pos” function is used in conjunction with “> 0” because, if “C” is present anywhere in the alpha string of “Q1”, it will return a value greater than zero. Otherwise, zero will be returned and “Q2” will not be asked.

Figure 6.12: Example of Code for an Alpha Character Search

```
Proc Q2
```

```
PreProc
```

```
ask if pos(“C”, Q1) > 0;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

6.7 Skips for Rosters

6.7.1 Setting Up Occurrence Labels

When creating a CAPI survey in CSPro, it is important that every roster has occurrence labels set (these are basically the names of each row in the roster). The reason the occurrence labels are important is that they will be displayed on the tablet screen to guide the interviewer.

To set the occurrence labels for a fixed roster (i.e., a roster where the rows are set and the names are known), right-click on the record in the data dictionary and select “Occurrence labels” (Figure 6.13).

The occurrence labels can then be entered according to the text in the questionnaire. The number of rows (5) would depend on the number set as “Max” in the records section of the data dictionary, as shown in Figure 6.14.

Figure 6.13: Setting Occurrence Labels for a Fixed Roster

N	Record Label	Record Name	Type Value	Require Max
	Household Demographics	SECTIONA	1	Yes 1
	Household Roster	SECTIONB	2	Yes 30
	Crop Roster	SECTIONC	3	Yes 5

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.14: Example of Occurrence Labels in a Fixed Roster

Occ	Label
1	Rice
2	Wheat
3	Potato
4	
5	

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

For a nonfixed roster, the names of the rows will not be known until the survey starts, so they will need to be dynamically determined by the system. For example, in the household roster, the occurrence label should be set to the name of each household member once the name is entered. This can be accomplished using the “setocclabel” function, as shown in Figure 6.15.

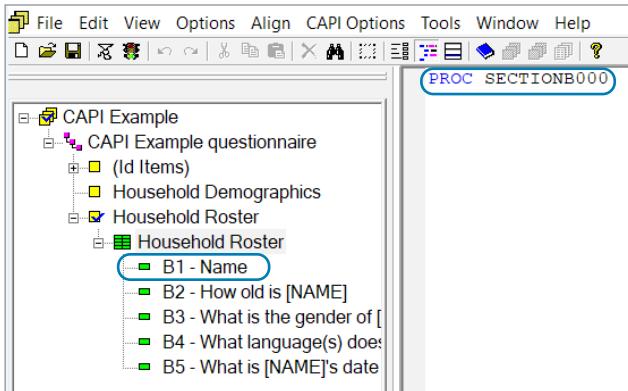
Figure 6.15: Dynamically Determining Occurrence Labels for Nonfixed Rosters

```
setocclabel(ROSTERNAME, OCCURRENCENAME);
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The roster name can be found by selecting it in the form viewer (Figure 6.16). Note that it (“SECTIONB000”) is different from the name given in the data dictionary.

Figure 6.16: Finding the Roster Name in the Form Viewer



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

For the code in Figure 6.17, the occurrence label is being set for the roster name “SECTIONB000” and the name for each row is from question “B1” (Name). The “strip” function is used to improve the

text formatting on screen, because the name variable is usually long (e.g., 30 or 50 characters) and this function removes all the excess spaces from the variable before setting it as the occurrence name. The strip function needs to be run after question “B1” is answered (“PostProc B1”) because the name value has then been stored and is ready to be used as the occurrence label.

Figure 6.17: Example of Code for Occurrence Labels in a Nonfixed Roster

```
Proc B1
setocclabel(SECTIONB000, strip(B1));
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

6.7.2 Ending a Nonfixed Roster

When designing a CAPI system, it is important to include a way to end a nonfixed roster and allow the interviewer to deal with the extra spaces or rows left in the data dictionary. Ideally, the roster should end without the interviewer having to press through all the empty rows. Below are two options for how to end a household roster.

Option 1: Controlling questions

A controlling question is a numeric question before the nonfixed roster. It is designed to capture how many occurrences the roster is going to have (Figure 6.18).

A controlling question must be created in a nonrostered section of the data dictionary. This numeric item can then be added to the form by dragging it from the dictionary to the top of the form that contains the roster (Figure 6.19). It should be unrostered because it is only asked one time before the roster begins.

Figure 6.18: Using a Controlling Question to End a Nonfixed Roster

How many people currently live in your household?
Numeric
"3"

Name	Age	Gender	Education	Language

Automatic stop at end of 3rd occurrence

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.19: Adding a Controlling Question to the Form

File Edit View Options Align CAPI Options Tools Window Help

CAPI Example
CAPI Example questionnaire
(Id Items)
Household Demographics
A10 - How many members are in y
Household Roster
Crop Roster

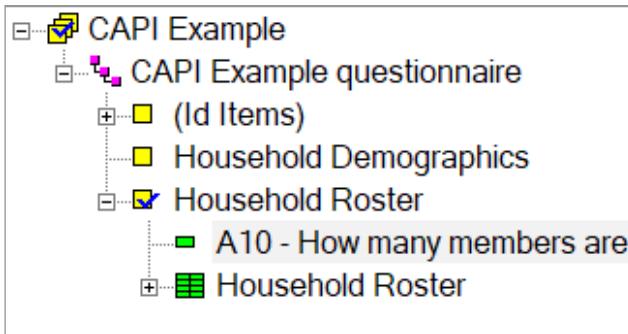
A10 - How many members are in your household?

	B1 - Name	B2 - How old is [NAME]	B3 - What is the gender of [NAME]
1			
2			
3			
4			
5			
6			
7			
8			
9			

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The controlling question appears above the roster on the right of screen, but its order still needs to be adjusted in the form tree or it will be asked after the roster. To do this, simply drag it above the roster on the form (Figure 6.20).

Figure 6.20: Adjusting the Order of a Controlling Question

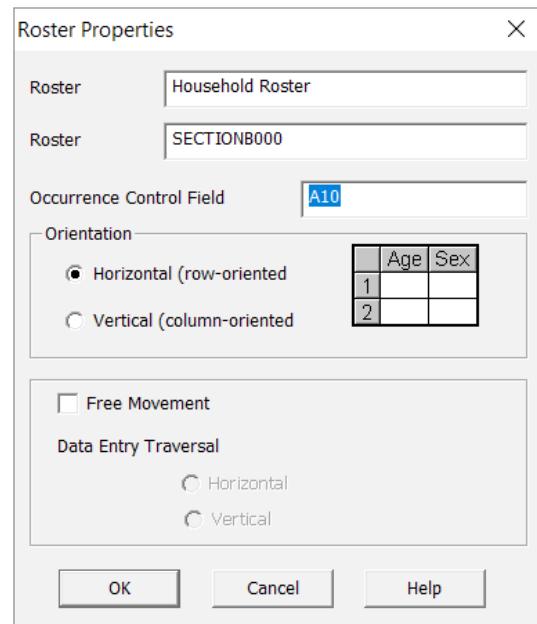


Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Next, right-click on the roster form and select “Roster Properties” (Figure 6.21).

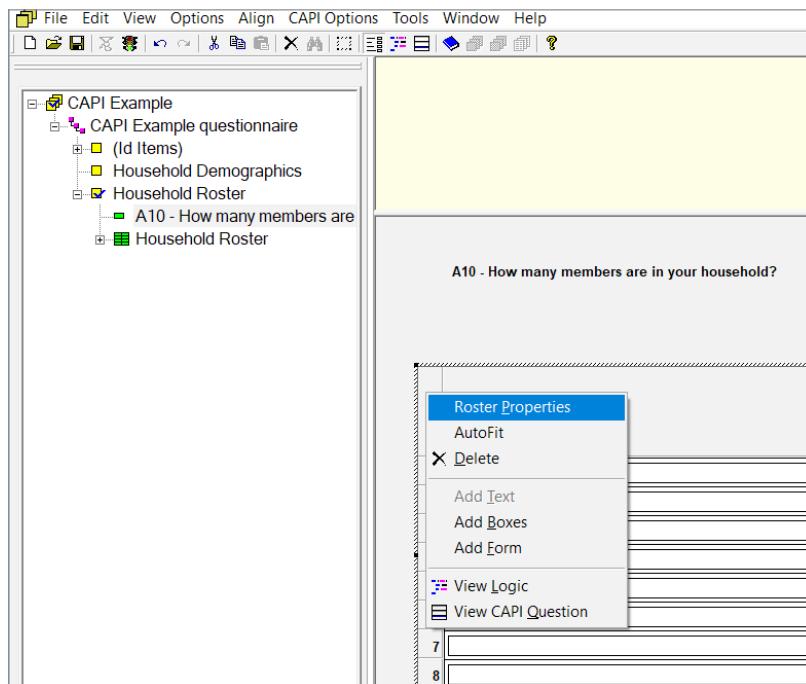
Finally, enter the item name that will control the number of occurrences and press “OK” (Figure 6.22).

Figure 6.22: Setting the Occurrence Control Field



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.21: Setting Roster Properties for a Controlling Question



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Option 2: Code using “endgroup”

Another way to end a nonfixed roster is to include logic code at the end of each occurrence, which will then check if that is the final occurrence (Figure 6.23).

To implement this in CSPro, an additional question needs to be added at the end of the record for the roster in the data dictionary, then inserted into the form (Figure 6.24).

Figure 6.23: Checking after Each Occurrence to End a Nonfixed Roster

Name	Age	Gender	Education	Language	More members?
Steve	35	M	12	English	Y
Jill	34	F	12	English	N

End of roster

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.24: Adding an Additional Question to End a Nonfixed Roster

The screenshot shows the CAPI Example questionnaire interface. On the left, the data dictionary is displayed in a tree view:

- CAPI Example
 - CAPI Example questionnaire
 - (Id Items)
 - Household Demographics
 - Household Roster
 - Household Roster
 - B1 - Name
 - B2 - How old is [NAME]
 - B3 - What is the gender of [NAME]
 - B4 - What language(s) does [NAME] speak?
 - B5 - What is [NAME]'s date of Birth?
 - B6 - Any more household members

On the right, the data entry form is shown. It consists of a table with 10 rows, each representing a household member. The columns are labeled:

- B1 - Name
- B2 - How old is [NAME]
- B3 - What is the gender of [NAME]
- B4 - What language(s) does [NAME] speak?
- B5 - What is [NAME]'s date of Birth?
- B6 - Any more household members

The first five columns have dropdown menus for their respective questions, while the last column has a dropdown menu for "Any more household members".

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The logic code would then need to be applied to the new question (“B6” in Figure 6.24). The logic works like the “skip to” function, except that the command “endgroup” is used in this case (Figure 6.25). This command ends the current roster and moves to the next part of the questionnaire.

Figure 6.25: Example of Logic Code to End a Nonfixed Roster

```
PROC B6
if $ = 0 then
    endgroup;
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

6.7.3 Skips Based on Answers in a Previous Roster

In some cases, the skip pattern for a question will be determined by the answers from an earlier roster. For instance, after the household roster is completed, there is an unrostered question that should be asked only for certain conditions. As an example, the question “Are any of the children in your household currently attending school?” should be skipped, unless there are children of school age in the household roster.

To check for this in the household roster, we would use the “count” function in conjunction with “ask if” as seen in Figure 6.26.

Figure 6.26: Basic Syntax to Check for a Condition in a Previous Roster

```
count(ROSTERNAME where CONDITION)
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

The “count” function will return a number for how many times the condition is satisfied in the roster. If it has not been satisfied, the function returns “0”. If, in the case of our example, there were any children of school age in the household, the count would be “1” or higher, so this needs to be included in the “ask if” statement (Figure 6.27).

Figure 6.27: Example of Code Using the Count Function

```
ask if count(SECTIONB000 where B2 in 4:18) > 0
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

6.7.4 Questions for Particular Occurrences

Sometimes in fixed rosters, there are questions that are only relevant to certain occurrences. A crop roster, for instance, may include rice and other crops, but rice is the crop of interest and this calls for additional questions pertaining only to rice. To ask such questions, the “curocc” function can be used. This function returns a numeric value for the current occurrence of the roster specified. If rice was the first occurrence in the roster, additional questions could be asked using the command in Figure 6.28.

Figure 6.28: Example of Code to Ask Questions by Occurrence

```
ask if curocc(SECTIONC000) = 1
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

6.8 Skip to Next

The “skip to next” command can be used to advance to the next row in a roster. It is used like a skip pattern to a subsequent question, but “next” is substituted for the variable’s name. This is useful for rosters with

an initial filter question, such as “Did you grow this crop in the last 12 months?”.

6.9 An Overview of Validations

Validations are vitally important when setting up a CAPI survey. They are the means of checking the validity of the answers recorded as responses are entered into the system. If an answer that is unexpected or impossible is entered by the interviewer, the CAPI system can be programmed to display a warning message for the interviewer to check and/or correct the value entered.

Real-time validations are a key benefit of using CAPI for fieldwork because the technology allows checks to take place immediately and when respondents are readily available for clarifications. Under the traditional pen and paper methodology, inconsistent or impossible answers would not be flagged until much later, when the paper forms undergo data entry and are checked on computers at headquarters.

As with skips, the logic applied to validations in CAPI can be much more complex and efficient than what is possible on paper forms. A survey with strong validations will result in higher-quality data being collected, as well as the associated benefit of vastly reduced data cleaning required after collection.

6.10 Hard and Soft Validations

In CSPro, two types of validations can be incorporated into questions. These are referred to as “hard” and “soft” validations. Essentially, a hard validation should be used when the answer is impossible, with the system not allowing the interviewer to continue past the question without correcting the value. A soft validation, on the other hand, should be used when

the answer is unexpected or possibly an entry mistake for a valid answer. In this instance, the system will display a warning message, allowing the interviewer to either move forward or correct the value entered.

To illustrate, let’s take the example of a question gathering the age of the household member. An age over 100 would be unusual, but it is possible and might be the correct age. It could also be an entry error, where the interviewer meant to type “10” and merely pressed an extra zero, resulting in a value of 100. Here, a soft validation can be used to have the interviewer check if the value entered is correct. However, there comes a point where an age value is impossible, say 120 years of age. Any value over 120 should therefore have a hard validation applied, so that the entry must be altered before the interviewer can proceed.

The basic syntax for a hard validation is shown in Figure 6.29. It needs to be run in the Proc section of the relevant question, or after the answer is entered. The code uses an “if” condition such that, when the condition is satisfied, an error message is shown and the only option provided is to reenter the value.

Figure 6.29: Basic Syntax for a Hard Validation

```
If [CONDITION] then
    errmsg ("the text to display in the error message is here");
    reenter;
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

If the question asking the ages of household members is “B2”, the syntax as shown in Figure 6.30 would be written on the logic interface in the form.

A soft validation uses similar syntax, but different options can be specified other than simply the instruction to reenter the value (Figure 6.31).

Figure 6.30: Example of Code for a Hard Validation

```
Proc B2
If $ > 119 then
    errmsg ("Age is too high please correct");
    reenter;
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.31: Basic Syntax for a Soft Validation

```
If [CONDITION] then
    errmsg ("the text to display in the error message is here")
    select ("first option text", ACTION,
            "second option text", ACTION);
endif;
```

ACTION can be a variable to move to or the special operator "continue" to move to next question.

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

To apply a soft validation in conjunction with the already defined hard validation in our example, the “if” condition should be applied only for values ranging from 100 to 119 (Figure 6.32). In these instances, the interviewer is given the option of reentering the age or moving to the next question.

Figure 6.32: Example of Code for a Soft Validation

```
Proc B2
If $ in 100:119 then
    errmsg ("Age is high please check")
    select ("Re-enter", B2,
            "Age is correct", continue);
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

6.11 Dynamic Text in Validation Error Messages

Error messages used in validations can be improved by inserting answers to previous questions into the message text. This can allow interviewers to more clearly understand the error at hand. We can use “%s” to pipe in an alpha value and “%d” to pipe in a numeric value (Figure 6.33).

Figure 6.33: Displaying Previous Answers in Error Messages

```
errmsg ("[NAME] is aged [AGE], is this correct?")
```

We can use %s to pipe in an alpha value

We can use %d to pipe in a numeric value

So our message will look like this.

```
errmsg ("%s is aged %d, is this correct?")
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Next, the sources of values for “%s” and “%d” need to be specified, as shown in Figure 6.34, with “B1” for name and “B2” for age. Further, the “strip” command is also used to remove the extra spaces from the household member’s name and the age entered.

Figure 6.34: Inserting Values for Previous Answers

```
errmsg("%s is aged %d, is this correct", strip(B1), B2)
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

6.12 Date Validations

Validations are essential for date questions. Basic operators can be used on these questions to make sure that dates are in the correct chronological relation to each other, e.g., that the date a child started school is after his or her date of birth.

A common date validation is to check that the respondent's date of birth matches with his or her age in years. To calculate this, the "datediff" function can be used to calculate the period between a respondent's date of birth and the interview date, to confirm whether this matches with the age in years.

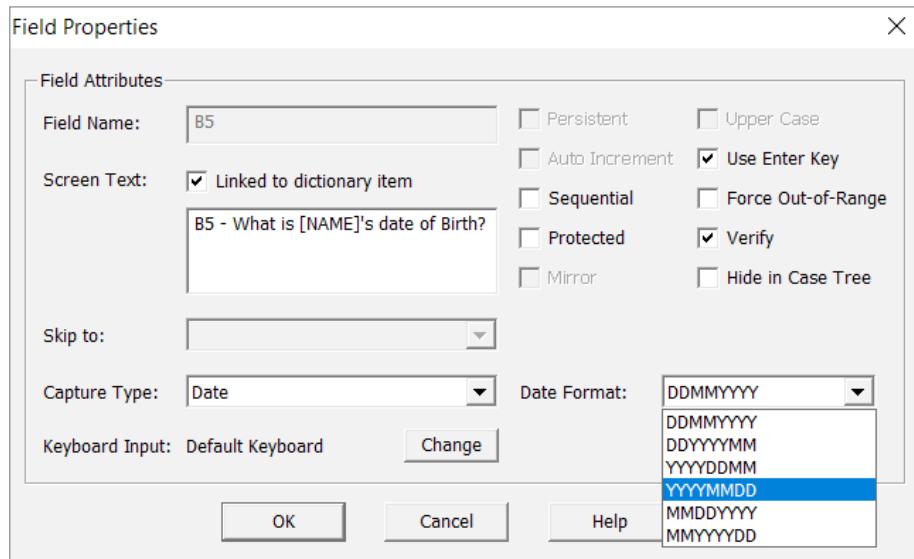
To use the "datediff" function, the date variables must be set to store the date in the "YYYYMMDD" format. To ensure this setting is correct, right-click on the field in the form, select "Field Properties", then change the date format in the dropdown menu (Figure 6.35).

Then the "datediff" function can be used to calculate the difference in years between the two date variables specified (EARLIERDATE and LATERDATE (Figure 6.36).

6.13 Multiple Value Sets

In some instances, valid responses to a single or multi select question will depend on an answer to a previous question. While validations could still be used to make sure that a legitimate answer has been selected, a more elegant way might be to use multiple

Figure 6.35: Setting the Date Format for an Age Validation



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.36: Basic Syntax to Calculate the Difference between Dates

`datediff(EARLIERDATE, LATERDATE, "difftype")`

Specifies how to return the difference e.g. "y" returns year integer. "d" days, "m" - default is days.

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

value sets. This way, only the valid answer options will be displayed on the screen, removing the need for validations.

Take the example of regional identifiers. In Figure 6.37, there are specific districts that should be displayed because they are within a province that has been previously selected. For example, if the Western Province is selected, only the districts of Colombo, Gampaha, and Kalutara should be displayed.

The setup for the province and district variables in the data dictionary should be a numeric data type, with value sets defined as in Figure 6.38: one value set containing the provinces and multiple value sets containing the districts in each province.

The final step is to write some logic code to tell the system when to display each value set. The “setvalueset” command can be run in PreProc, so the value sets are loaded as the question displays to the interviewer, and the “elseif” function can be repeated as many times as required (Figure 6.39).

Figure 6.37: Example of Regional Identifiers for Use in Multiple Value Sets

A	B	C	D
1 Province Name	Province Code	District Name	District Code
2 Western		1 Colombo	11
3 Western		1 Gampaha	12
4 Western		1 Kalutara	13
5 Central		2 Kandy	21
6 Central		2 Matale	22
7 Central		2 Nuwara Eliya	23
8 Southern		3 Galle	31
9 Southern		3 Matara	32
10 Southern		3 Hambantota	33
11 Northern		4 Jaffna	41
12 Northern		4 Mannar	42
13 Northern		4 Vavuniya	43
14 Northern		4 Mullaitivu	44
15 Northern		4 Kilinochchi	45
16 Eastern		5 Batticaloa	51
17 Eastern		5 Ampara	52
18 Eastern		5 Trincomalee	53
19 North Western		6 Kurunegala	61
20 North Western		6 Puttalam	62
21 North Central		7 Anuradhapura	71
22 North Central		7 Polonnaruwa	72
23 Uva		8 Badulla	81
24 Uva		8 Moneragala	82
25 Sabaragamuwa		9 Ratnapura	91
26 Sabaragamuwa		9 Kegalle	92

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.38: Example of Multiple Value Sets for District

N	Value Set Label	Value Set Nar Value Label	From	To	Special	Image
	A01 - District	A01_DIS_VS9				
		Ratnapura	91			...
		Kegalle	92			...
	A01 - District	A01_DIS_VS8				
		Badulla	81			...
		Moneragala	82			...
	A01 - District	A01_DIS_VS7				
		Anuradhapura	71			...
		Polonnaruwa	72			...
	A01 - District	A01_DIS_VS6				
		Kurunegala	61			...
		Puttalam	62			...
	A01 - District	A01_DIS_VS5				
		Batticaloa	51			...
		Ampara	52			...
		Trincomalee	53			...
	A01 - District	A01_DIS_VS4				
		Jaffna	41			...
		Mannar	42			...
		Vavuniya	43			...
		Mullaitivu	44			...
		Kilinochchi	45			...
	A01 - District	A01_DIS_VS3				
		Galle	31			...
		Matara	32			...
		Hambantota	33			...
	A01 - District	A01_DIS_VS2				
		Kandy	21			...
		Matale	22			...
		Nuwara Eliya	23			...
	A01 - District	A01_DIS_VS1				
		Colombo	11			...
		Gampaha	12			...
		Kalutara	13			...

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 6.39: Example of Code to Display Answer Options by Value Set

```
PreProc  
if A01 = 1 then  
    setvalueset ($,A01_DIS_VS1);  
elseif A01 = 2 then  
    setvalueset ($,A01_DIS_VS2);  
endif;
```

If further regional identifiers (such as those for village or primary sampling unit) are required for the survey, the process described here can be duplicated using the relevant value sets.

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 7: Surveys Conducted in Multiple Languages

Many CAPI surveys will require more than one language for fieldwork. To add additional languages in CSPro, a version of the alternative language questionnaire should be created in Microsoft Word or Excel and should match with the master language version. The additional languages can then be copied from Word or Excel and pasted into CSPro in the relevant places, so that they can be seen on the tablets by the interviewers.

In CSPro, the additional languages need to be inserted into three separate parts of the program: the data dictionary, the question text, and the error messages. Figure 7.1 shows some of the areas of the tablet screen for which alternative language text will be required.

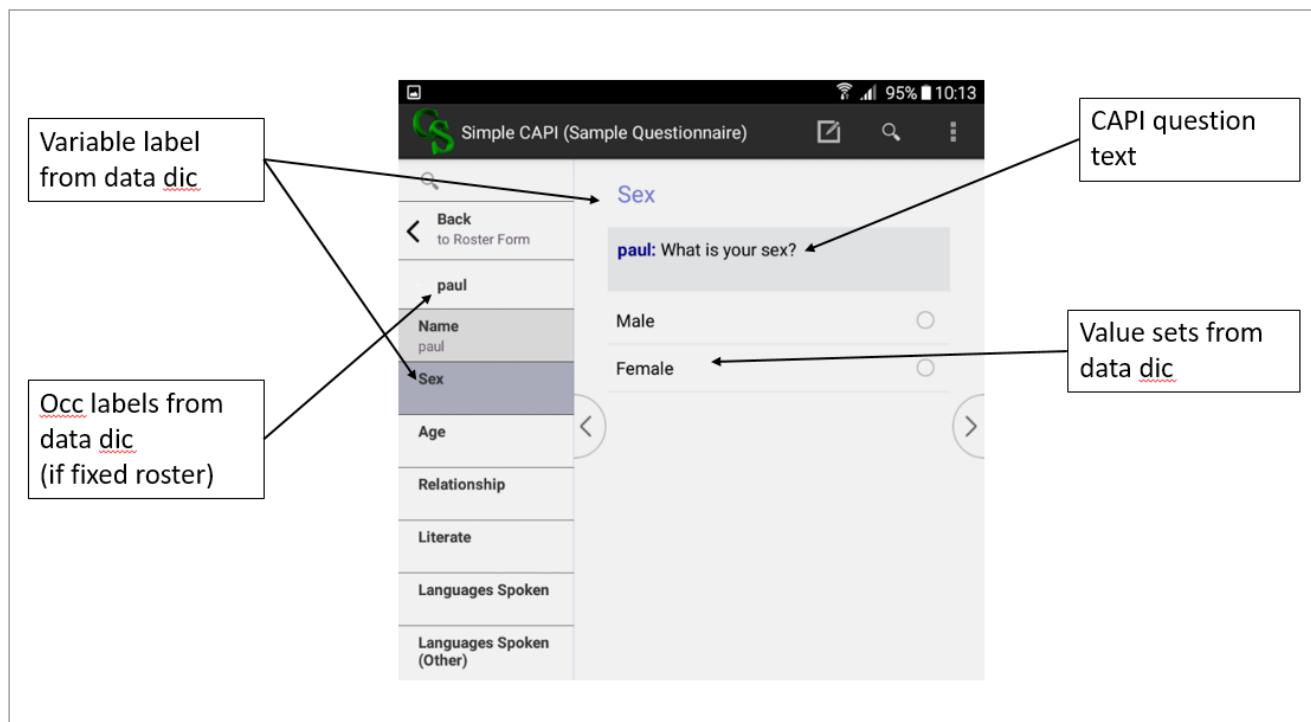
7.1 Alternative Languages in the Data Dictionary

Any additional language required needs to be added to the data dictionary. This can be done by selecting “Languages” from the edit menu (Figure 7.2).

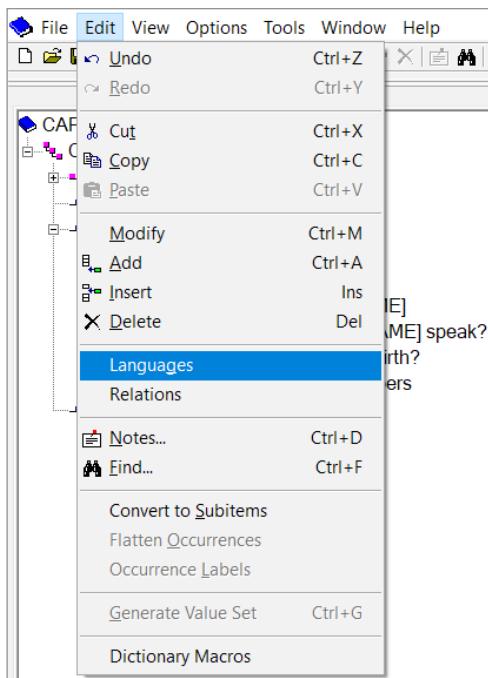
The new language can be added by entering a two-letter code and the name of the language (Figure 7.3).

Next, a dropdown menu will present the languages available for the questionnaire (Figure 7.4). Select the newly added language from the options in the dropdown.

Figure 7.1: Example of Screen Areas to Display Additional Languages



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.2: Accessing the Language Options in the Data Dictionary

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Once the language has been changed, the alternative language can be pasted over the top of the original language (Figure 7.5). The item name (e.g., “B1”) and variable (e.g., “NAME”) can remain in English.

This process needs to be repeated for all value sets (Figure 7.6) as well as for occurrence labels in fixed rosters (Figure 7.7).

7.2 Alternative Languages in Question Text

Any additional languages will need to be applied to the question text set in CSPro. To add a language in this section, go to the “CAPI Options” menu and select “Define CAPI Languages” (Figure 7.8).

Figure 7.3: Adding a New Language in the Data Dictionary

Languages		
	Add	Remove
1	EN	English
2	KH	Khmer

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.4: Selecting an Alternative Language for the Questionnaire

The screenshot shows the CSPro questionnaire editor interface. On the left, there is a tree view of the questionnaire structure under 'CAPI Example questionnaire'. On the right, there is a table with columns for 'Item Label' and '(record type)'. A dropdown menu labeled 'Language' is open, showing options: EN: English, EN: English, and KH: Khmer. The 'KH: Khmer' option is highlighted with a blue selection bar. The table rows include: N Item Label (record type), Village, Household ID, B1 - Name, B2 - How old is [NAME], B3 - What is the gender of [NAME], B4 - What language(s) does [NAME] speak?, B5 - What is [NAME]'s date of Birth?, and B6 - Any more household members.

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.5: Adding the Alternative Language to Item Labels

The screenshot shows the CSPro software interface with the 'Language' dropdown set to 'KH: Khmer'. On the left, a tree view shows a project structure under 'CAPI Example' with 'Household Demographics' and 'Household Roster' expanded. The 'Household Roster' node has several child items, some of which are selected and highlighted in blue. On the right, a table titled 'Item Label' lists these items with their English names and Khmer labels.

N	Item Label (record type)	Item Name
	Village	VILLAGE
	Household ID	HOUSEHOLD_ID
	B1 - ឈ្មោះ	B1
	B2 - ការពិពណ៌នា [NAME]	B2
	B3 - What is the gender of [NAME]	B3
	B4 - What language(s) does [NAME] speak?	B4
	B5 - What is [NAME]'s date of Birth?	B5
	B6 - Any more household members	B6

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.6: Adding the Alternative Language to Value Sets

The screenshot shows the CSPro software interface with the 'Language' dropdown set to 'KH: Khmer'. On the left, a tree view shows a project structure under 'CAPI Example' with 'Household Demographics' and 'Household Roster' expanded. The 'Household Roster' node has several child items, some of which are selected and highlighted in blue. On the right, a table titled 'Value Set Label' lists these items with their English labels and Khmer labels.

N	Value Set Label	Value Set Nar	Value Label	From
	B3 - What is the gender of [NAME]	B3_VS1	ធម្ម	1
			ឬស	2

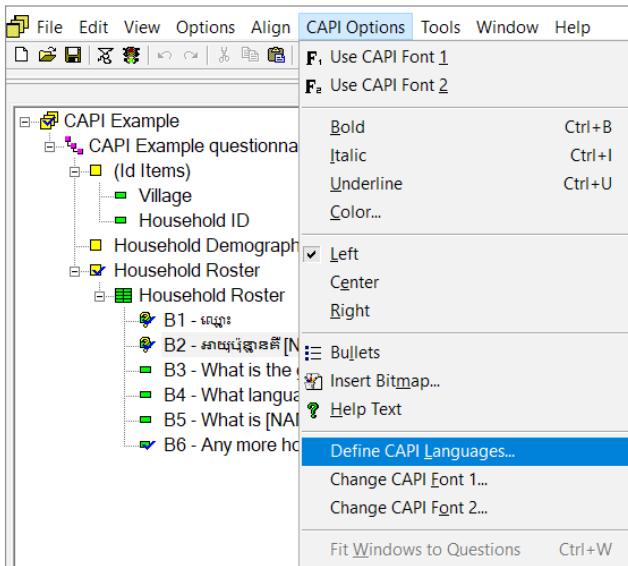
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.7: Adding the Alternative Language to Occurrence Labels

The screenshot shows the CSPro software interface with the 'Language' dropdown set to 'KH: Khmer'. A modal dialog box titled 'Occurrence Labels (Crop Roster)' is open, displaying a table with two columns: 'Occ' and 'Label'. The table lists five entries, each with an English label and its Khmer translation.

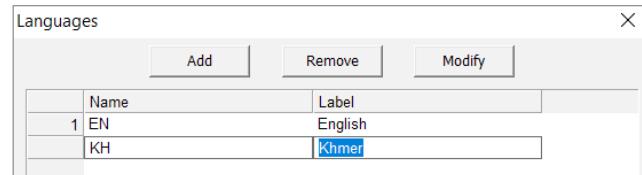
Occ	Label
1	អង្គ
2	តំបន់
3	Carrot
4	Cassava
5	Wheat

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 7.8: Defining Alternative Languages for Question Text

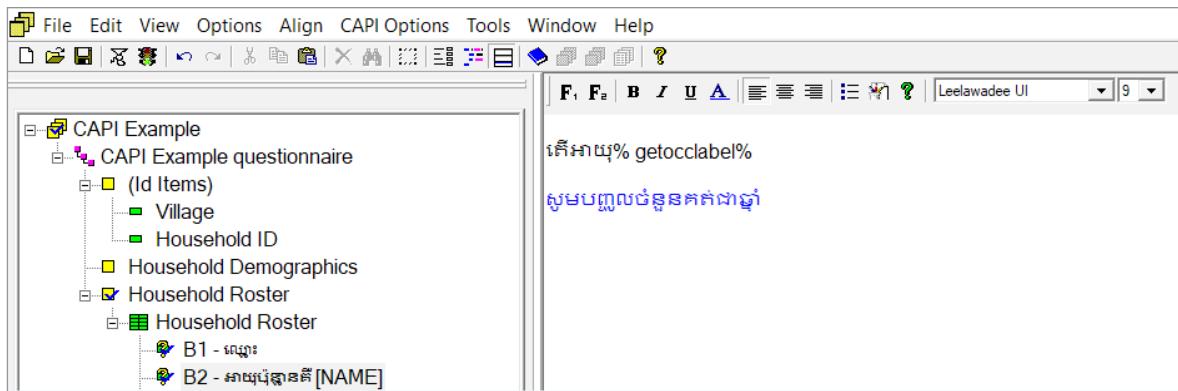
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

The additional language can now be added (Figure 7.9). It is important that the language's two-letter code and its name match those added in the data dictionary.

Figure 7.9: Adding the Alternative Language for Question Text

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

After the language has been added, the question text and interviewer instructions can be converted into the alternative language. If you are using text piping, such as "%getocclabel%", this code needs to remain in English and be inserted into the text in the relevant grammatical place (Figure 7.10).

Figure 7.10: Example of Text Piping within Alternative Language Text

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

7.3 Alternative Languages in Error Messages

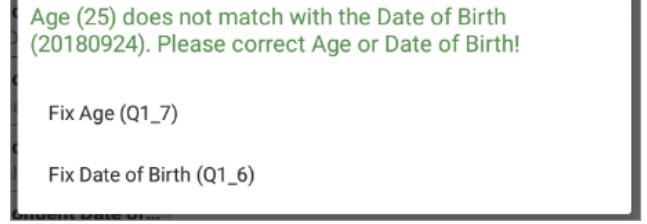
The questionnaire's error messages also need to be updated to reflect any additional languages required for the survey. The "getlanguage" function is useful for this because it fetches the language currently being used by the CAPI system and returns it as a two-letter string (e.g., "EN" or "KH"). A nested "if" statement inside the logic code for each error message, as in Figure 7.11, can be used to ensure that an error message will be displayed in the correct language (Figure 7.12).

Figure 7.11: Basic Syntax to Display an Alternative Language in an Error Message

```
if getlanguage() = "EN" then  
    errmsg("Hello");  
elseif getlanguage = "KH" then  
    errmsg("សូមបញ្ជីតាមការបង្កើតរបស់អ្នក");  
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 7.12: Example of an Error Message Displayed on a Tablet



Age (25) does not match with the Date of Birth (20180924). Please correct Age or Date of Birth!

Fix Age (Q1_7)

Fix Date of Birth (Q1_6)

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 8: Setting Up the Completed Questionnaire for Data Collection

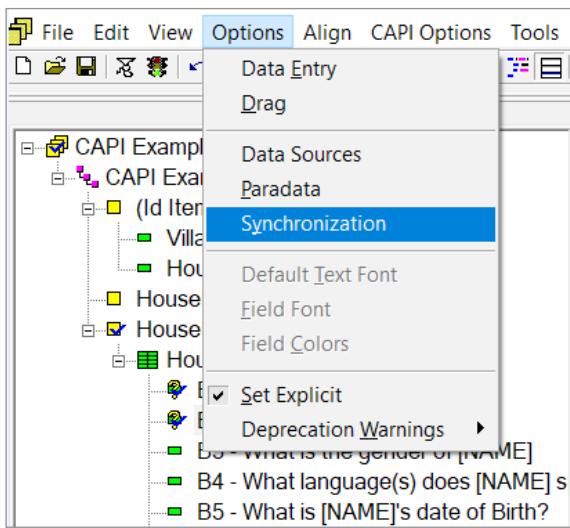
Once the CAPI questionnaire has been created in CSPro—including completion of the data dictionary, forms, question text, skips, validations, and additional languages—it needs to be configured for data collection. This involves setting up the syncing protocols, so that the system knows where to send completed cases and other settings to optimize the questionnaire for CAPI use.

After this, tablets or other handheld digital devices can be set up by installing the CSEntry app, then deploying the CAPI questionnaire to the app. Note that it is recommended to install the app and deploy the questionnaire to only one tablet initially, then to test thoroughly before proceeding with installation and deployment to all fieldwork tablets.

8.1 Setting the Syncing Protocols in CSPro

To set up syncing protocols, go to the options menu in CSPro and select “Synchronization” as shown in Figure 8.1.

Figure 8.1: Accessing the Syncing Options in CSPro

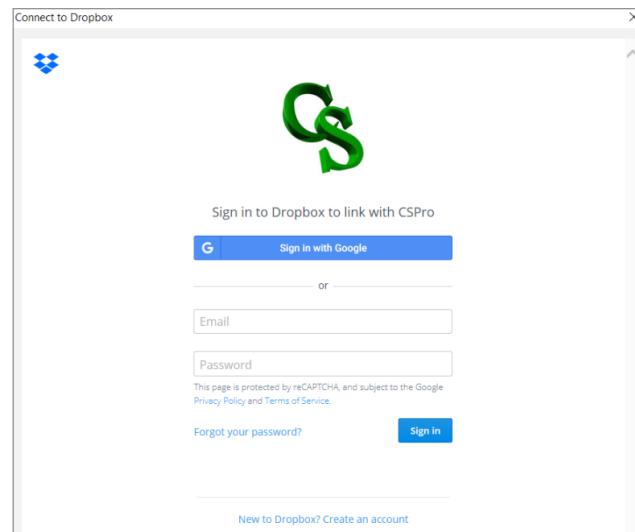


Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Next, select the server of choice. You will need to have already set up the server, either by registering for an account at Dropbox or using your own local or cloud-based server via CSWeb or FTP (section 2.4).

First, select the server type, then click “Test connection”. A pop-up window will then display to sign in and set a successful connection to the server type (Figure 8.2).

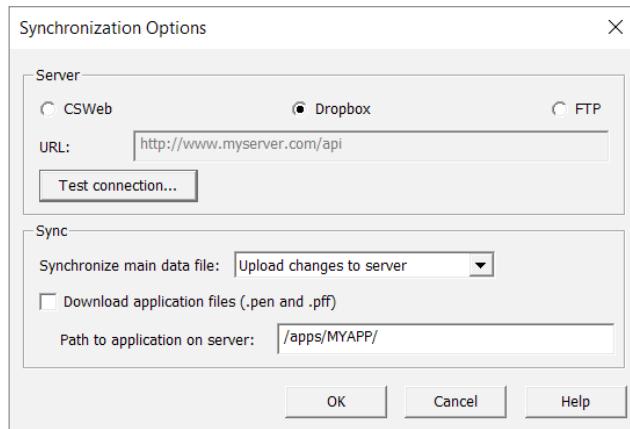
Figure 8.2: Example of Signing In to a Server (Dropbox)



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

After this, the sync settings can be configured using the options shown in Figure 8.3, as follows:

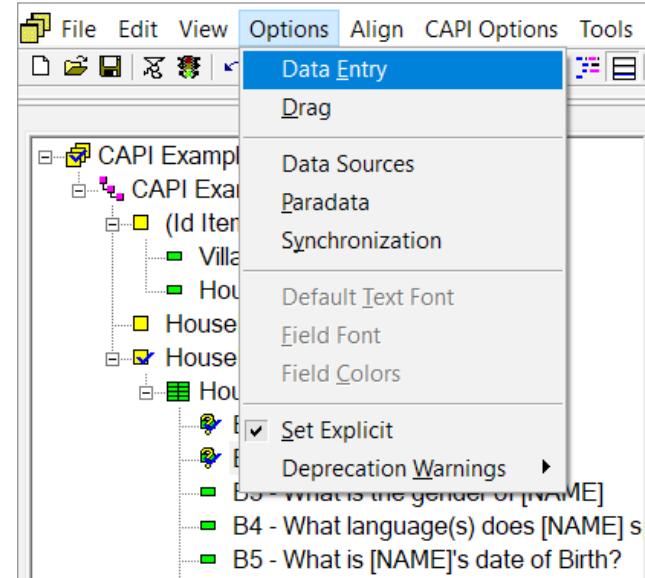
“Synchronize main data file”. The default option, “Upload changes to server”, will send any newly completed survey cases to the server. The “Download changes to server” option can be used by supervisors to download all cases from the server to their tablets. The final option “Sync local and remote changes” will do both. For a basic CAPI project, it is recommended to use only the first option, “Upload changes to server”.

Figure 8.3: Establishing the Syncing Protocols

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

“Download application files”. This option can be selected if a user requires the capacity to make changes to the system via the server during fieldwork. It is recommended to select this option, even if changes are not planned (as they may be required at a later date).

there are some preferences under “Mobile Options” that will change how the CAPI system is displayed on the tablets.

Figure 8.4: Accessing the Data Entry Options in CSPro

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

8.2 Refining the Data Collection Options in CSPro

Because CSPro was initially designed for desktop data entry, there are a few settings that need to be refined to optimize a survey for CAPI. For instance, it is very important to configure the system to allow for partial data entry. This is because, during fieldwork, there are instances where an interview might be interrupted and stopped part way through. In such cases, the respondent might ask the interviewer to come back to complete the questionnaire at a later time, or the questionnaire may remain partially completed. To allow for partial entry, press the options menu in CSPro, then select “Data Entry” (Figure 8.4).

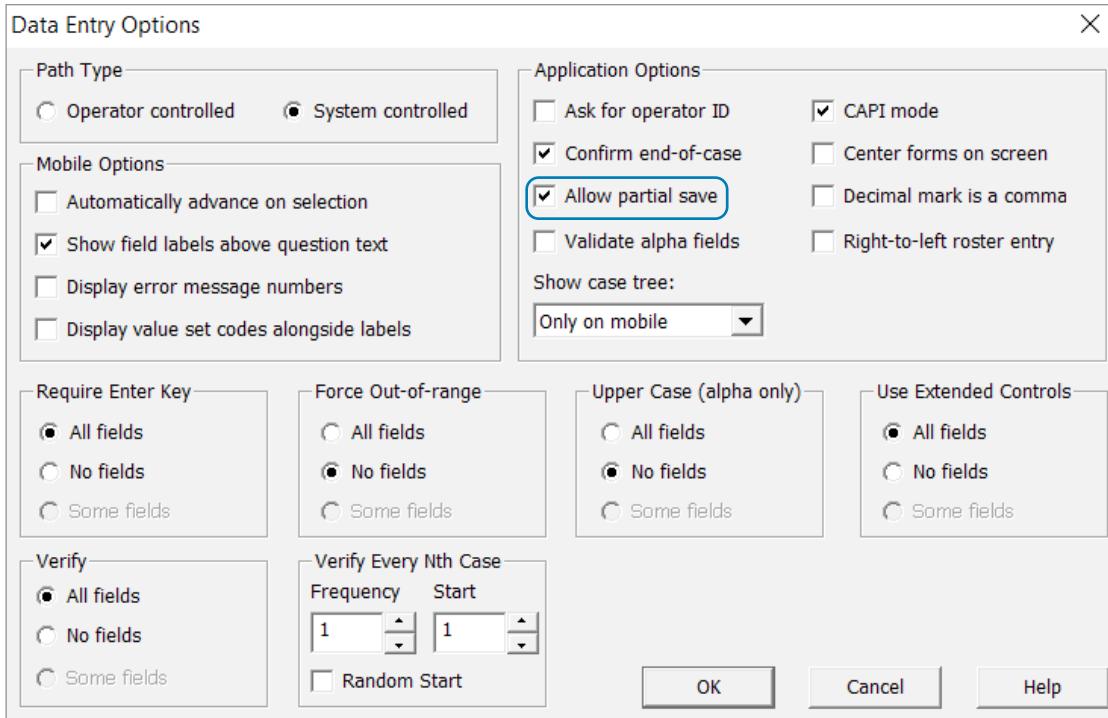
Next, ensure that the checkbox for “Allow partial save” is ticked (Figure 8.5). On this screen,

8.3 Installing CSEntry onto Tablets

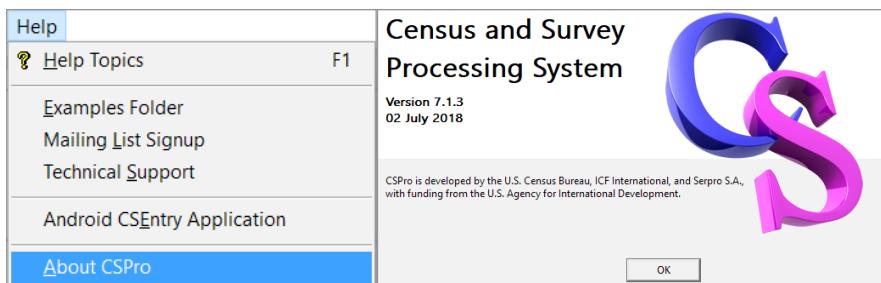
There are two ways to install CSEntry onto tablets:

- Connect to a Google Play account, then download and install the app, on each tablet
- Install manually using an Android package (.apk) file

We will use the second method, so that we do not need to set up or sign in to a Google Play account on every tablet. To find the “.apk” file required, simply conduct a search using Google. Keep in mind that the version of CSEntry you install onto the tablets must always match the version of CSPro being used on the PC. To determine the version of the latter, click on the help menu in CSPro, then select “About CSPro” (Figure 8.6).

Figure 8.5: Allowing for Saving of Partially Completed Questionnaires

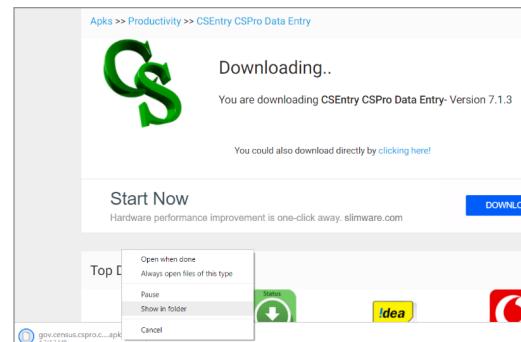
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 8.6: Determining the CSPro Version in Use

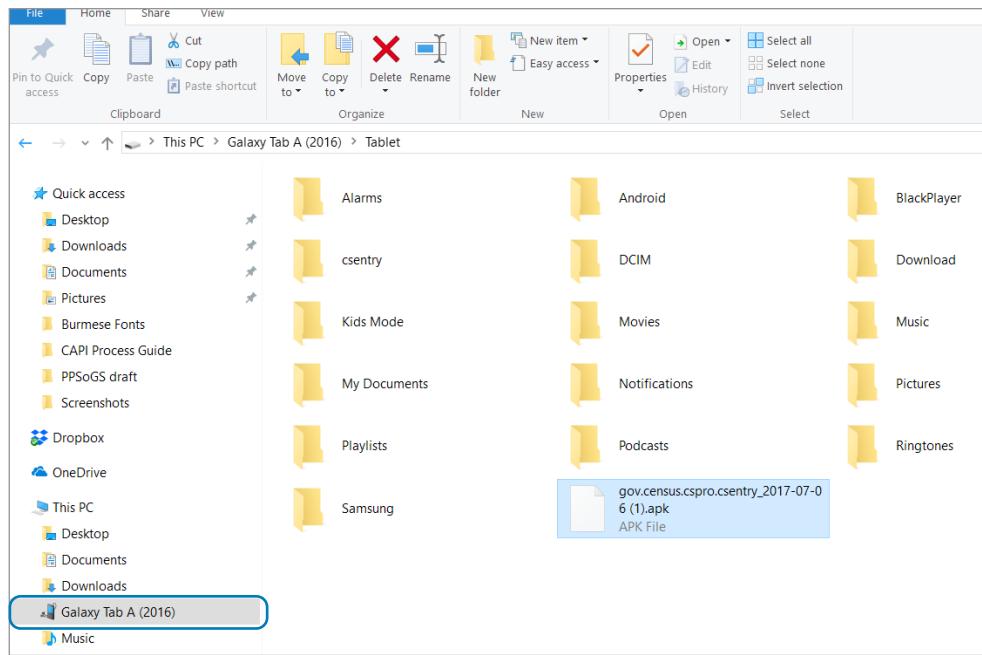
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Next, use Google to search for “CSEntry .apk”. There are several websites that will have the up-to-date “.apk” file. Once the file finishes downloading, right-click on it, then select “Show in folder” (Figure 8.7).

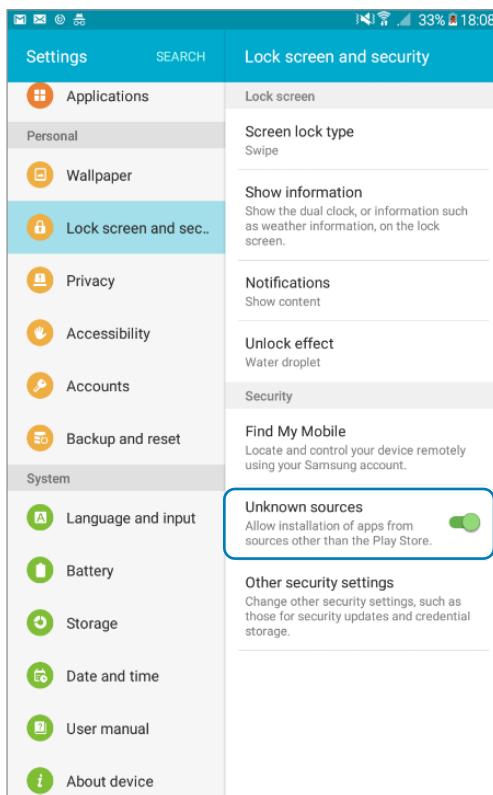
Next, using a USB cable, copy the downloaded install file onto the device storage of the tablet (Figure 8.8).

Figure 8.7: Downloading the CSEntry Install File

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 8.8: Copying the CSEntry Install File to the Tablet

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 8.9: Allowing Installation of Apps from Unknown Sources

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

On the tablet, go to the settings menu, select “Lock screen and security”, navigate down to “Unknown sources”, and ensure the toggle button is to the right (Figure 8.9).

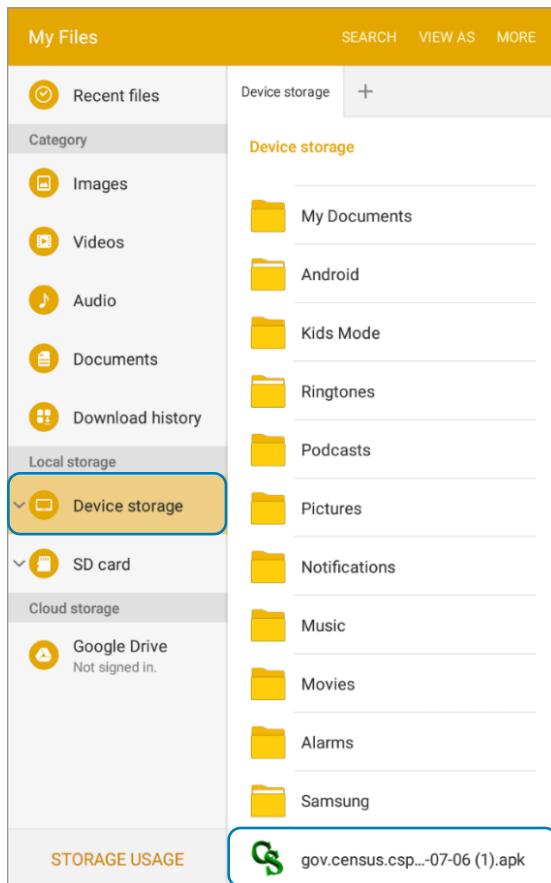
Now, on the home screen of the tablet, press “My Files” (Figure 8.10).

Figure 8.10: Accessing the Install File on the Tablet

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

On the left of the next screen, press “Device storage”, then press on the “.apk” file to install (Figure 8.11).

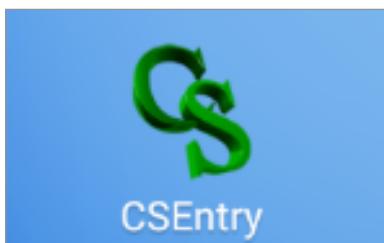
Figure 8.11: Initiating the Install File via Device Storage



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

After the installation, the CSEntry icon will appear among the tablet’s apps on the home screen (Figure 8.12).

Figure 8.12: CSEntry Icon on Tablet



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

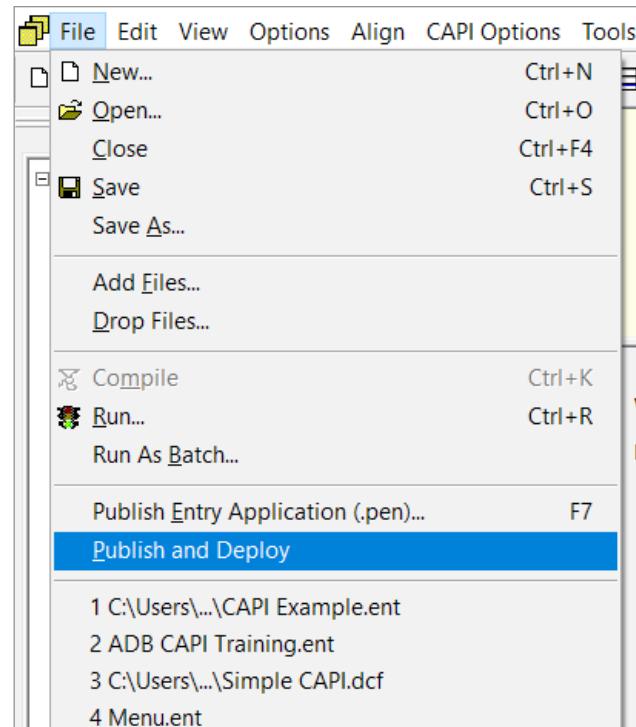
8.4 Deploying the Questionnaire to the Tablets

The CAPI questionnaire can be deployed directly onto the tablet by connecting to the PC using a USB cable. The questionnaire can also be uploaded to Dropbox, then deployed to each tablet from there via the internet.

8.4.1 Deploying Directly to the Tablet

First, using a USB cable, connect the tablet to the PC. Then, with the desired questionnaire open, from the file menu in CSPro, select “Publish and Deploy” (Figure 8.13).

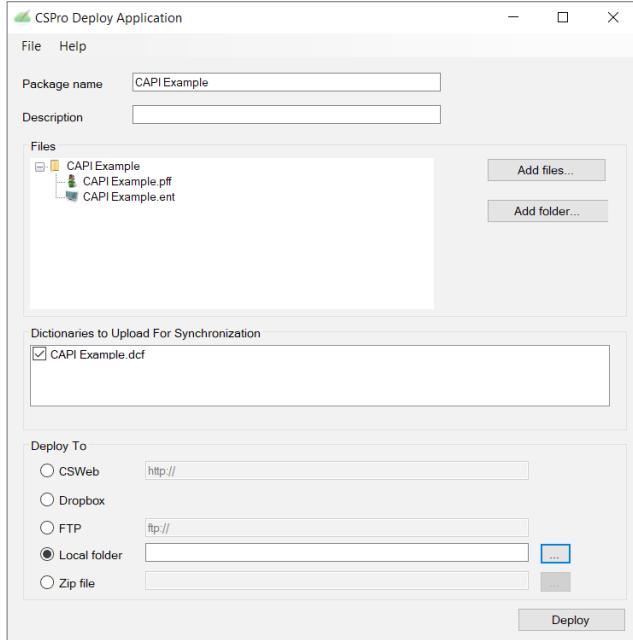
Figure 8.13: Initiating Deployment of the Questionnaire from CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

After this, you may add any additional files that may be required for the survey (e.g., lookup tables) as shown in Figure 8.14. Next, under the “Deploy To” options, select “Local folder” and specify the destination folder by clicking on the button on the right.

Figure 8.14: Specifying the Deployment Options in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

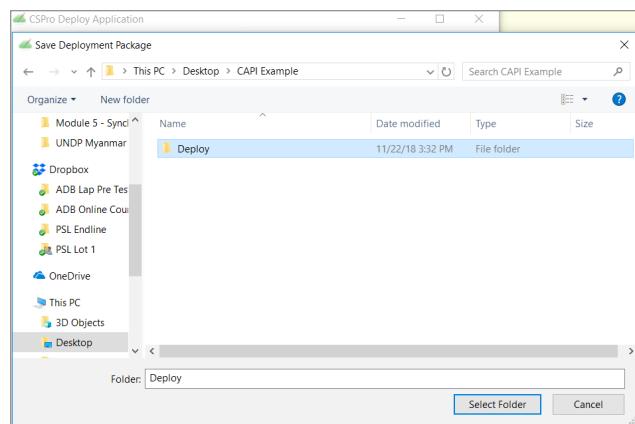
Create or select a local folder on the computer to store the deployment files (Figure 8.15). then back at the “Deployment Application” screen, press “Deploy”.

The files saved to the “Deploy” folder are the ones you can copy across to the tablet, transferring them into the appropriate destination folder (e.g., “CAPI Example”) using Windows File Explorer (Figure 8.16).

8.4.2 Deploying via Dropbox

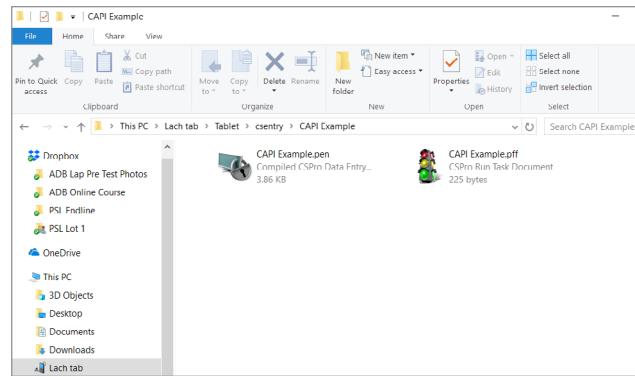
To deploy the questionnaire to the tablets via Dropbox, follow the initial steps in section 8.4.1, but

Figure 8.15: Creating the Destination Folder for the Deployment Files



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 8.16: Questionnaire Files Deployed to Tablet

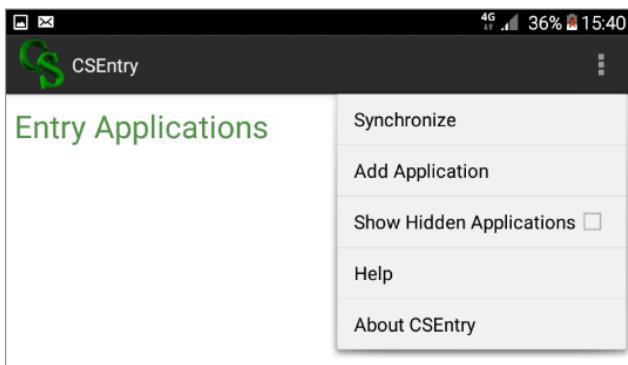


Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

select “Dropbox” when you arrive at the “Deploy To” options. The system will deploy the questionnaire to the same Dropbox account that was specified in the sync settings (section 8.1).

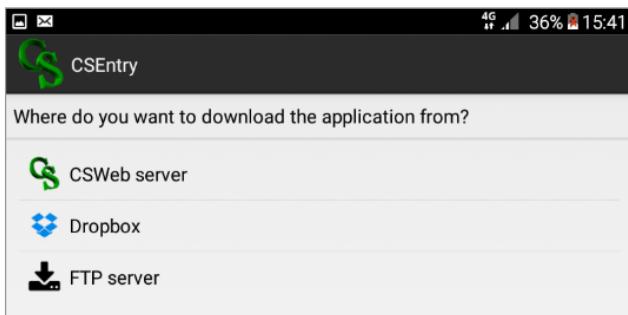
You are now ready to deploy the questionnaire to each tablet.

On the tablet, open the CSEntry app, press on the menu icon at the top right of screen, then press “Add Application” (Figure 8.17).

Figure 8.17: Preparing to Import the Questionnaire into the Tablet

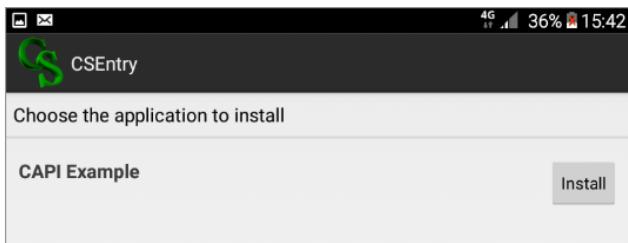
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

From the list of application sources, select “Dropbox” (Figure 8.18).

Figure 8.18: Selecting the Source to Import the Questionnaire

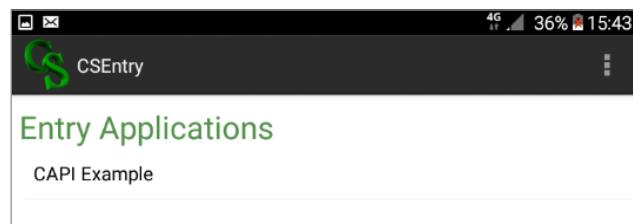
Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

CSEntry will connect to Dropbox and display a list of all the CAPI questionnaires available. Select the relevant questionnaire and press the install button (Figure 8.19).

Figure 8.19: Installing the Questionnaire onto the Tablet

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

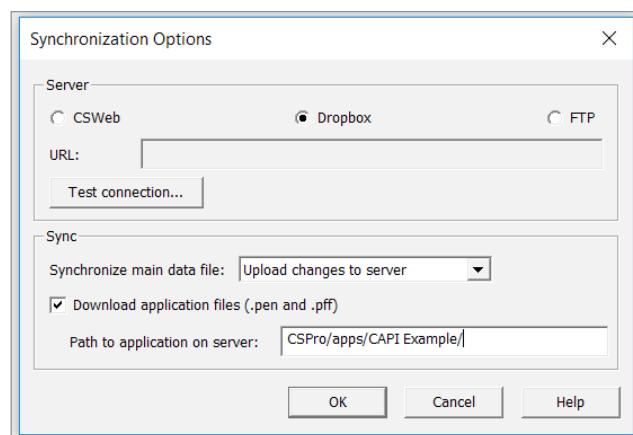
The questionnaire will then be downloaded and set up for use (Figure 8.20).

Figure 8.20: Questionnaire Ready for Use on Tablet

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

8.5 Questionnaire Updates via the Server

If questionnaire or CAPI system updates are required during fieldwork (or at any point after the initial deployment), you will need to manually upload the relevant files to the server, specifying the folder that was created during the establishment of the syncing protocols (Figure 8.21).

Figure 8.21: Syncing Options for Connecting to the Server (Dropbox)

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

To upload the necessary update files, sign in to Dropbox (or other server), navigate to the folder for the current project, and replace the .pff and .pen files of the same name (Figure 8.22).

Figure 8.22: Overwriting the Questionnaire Files on the Server (Dropbox)



The screenshot shows a Dropbox interface with a sidebar on the left containing 'My files', 'Sharing', 'File requests', and 'Deleted files'. The main area displays a folder named 'CAPI Example' with two files listed: 'CAPI Example.pen' and 'CAPI Example.pff'. Both files were modified 11 seconds ago and are owned by 'Only you'.

Name	Modified	Members
CAPI Example.pen	11 secs ago	Only you
CAPI Example.pff	9 secs ago	Only you

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 9: Quality Control during Data Collection

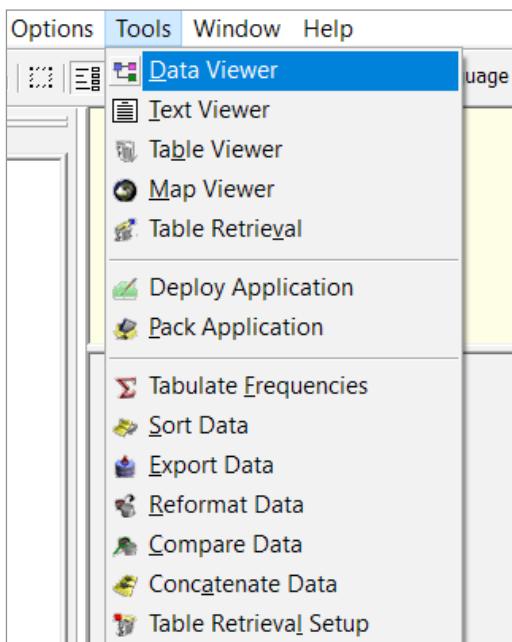
During fieldwork, there are several features of CSPro that can be used to improve the quality of the data collected. The features utilized will depend on the individual circumstances of each CAPI project, such as survey complexity and budget. Nonetheless, it is recommended to use as many features as possible to ensure the best-quality data is collected.

9.1 Downloading Data from the Server

Once cases are completed and uploaded during fieldwork, they can then be downloaded from the server in CSPro data base format (.csdb). This format can be converted and exported to other formats such as Microsoft Excel, Statistical Package for the Social Sciences (SPSS), STATA, etc. using the “Export Data” tool in CSPro.

Data can be downloaded in CSPro by navigating to the tools menu and opening the Data Viewer (Figure 9.1).

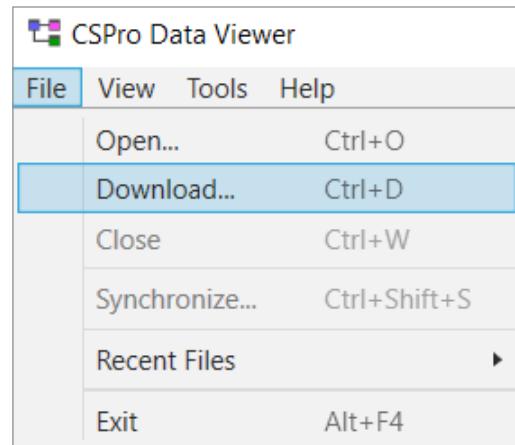
Figure 9.1: Opening the Data Viewer in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

In the Data Viewer, select “Download” from the file menu (Figure 9.2).

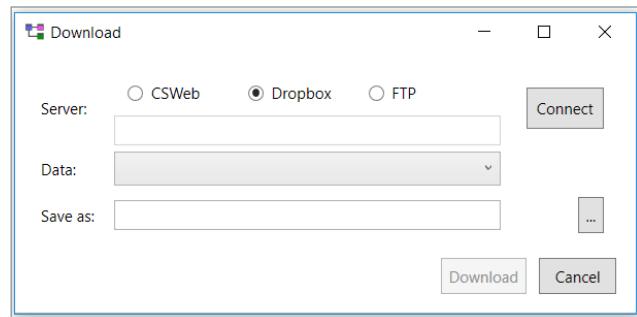
Figure 9.2: Initiating the Download of a Data File



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

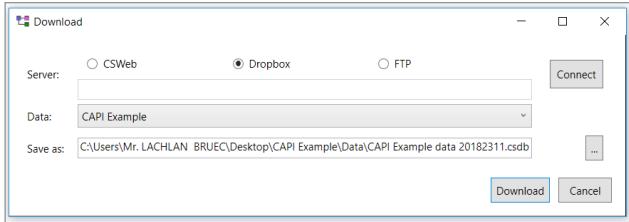
On the “Download” dialogue box, select the server type and click “Connect” (Figure 9.3).

Figure 9.3: Specifying the Server to Retrieve the Data File



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

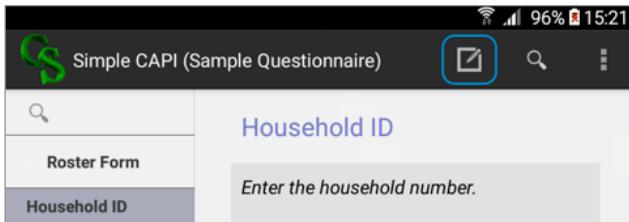
Once connected, select the data file (in the case of more than one questionnaire hosted on this server), then specify a destination folder and filename for the file (Figure 9.4). It is recommended to have already created a folder or filename that includes the date of the download, since the data will be exported often. Finally, click on “Download”.

Figure 9.4: Downloading the Data File to the Local Directory

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

9.2 Making and Viewing Survey Notes

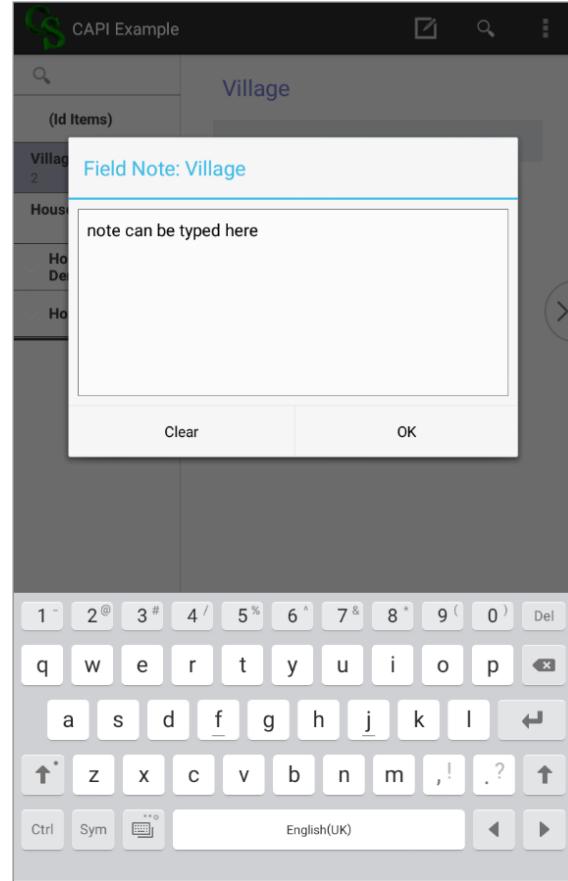
Similar to the traditional pen and paper method, CSPro allows interviewers to make field notes on any question in a CAPI survey. Doing so is sometimes necessary to further explain an answer or provide context. Interviewers can make notes for any given question by simply pressing on the edit icon at the top of the CSEntry screen, as shown in Figure 9.5.

Figure 9.5: Accessing the Field Note Functionality in CSEntry

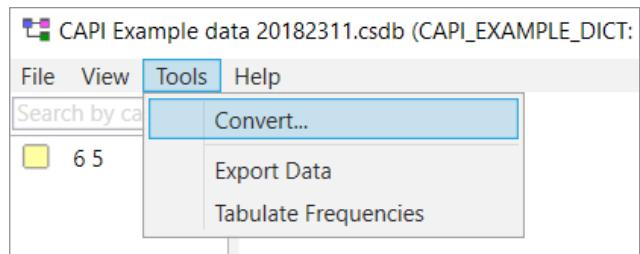
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

This then opens a text box to allow the interviewer to enter a note about the question on screen (Figure 9.6).

Notes from fieldwork can be viewed by users at survey headquarters by downloading the data into the CSPro Data Viewer. Since the data will need to be converted to a text file, select “Convert” from the tools menu (Figure 9.7).

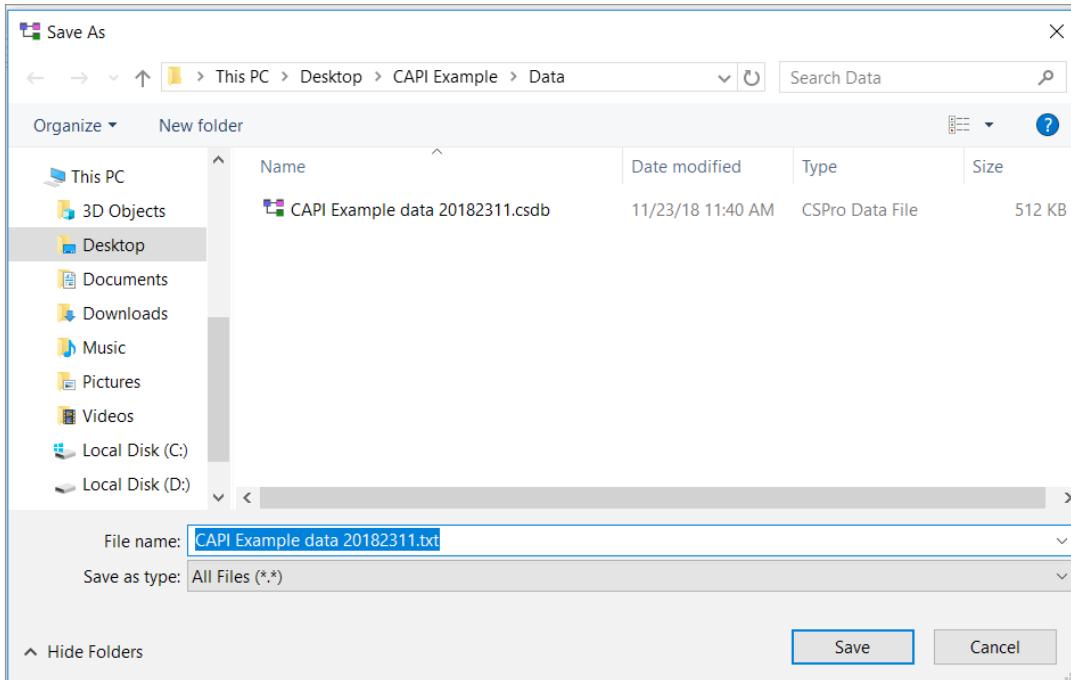
Figure 9.6: Entering a Field Note on the Current Question

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.7: Converting the Data File to a Text File

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Select “All files” in the dropdown menu at the base of screen, type your preferred filename and end it with a “.txt” extension, then click “Save” (Figure 9.8).

Figure 9.8: Saving the Text File to the Local Directory

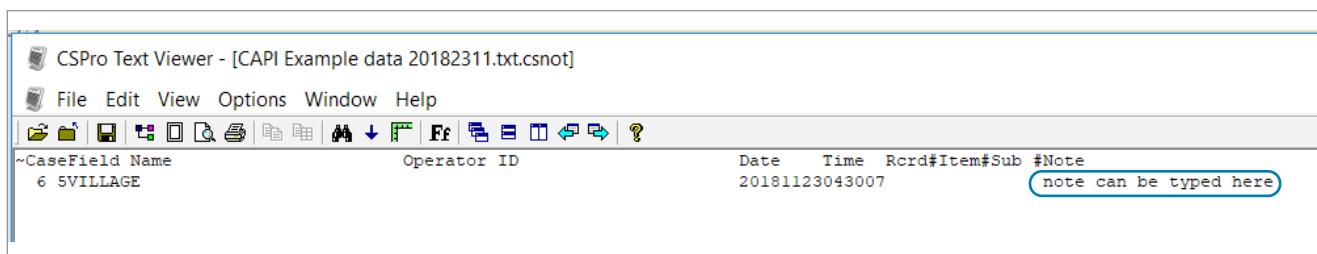
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Once converted, the text file and some others will be created. A file ending with the extension “.csnot” will also be created. This file can be viewed in the CSPro Text Viewer, allowing users to see any comments left by interviewers, with the relevant case and question referenced (Figure 9.9).

9.3 Quality Control Tables

A CSPro tabulation function can create tables that can be run during fieldwork for quality control

purposes. Such tables can be designed with two functions in mind. They can be used to check the progress of fieldwork, such as the number of completed interviews of each status, team, or region. They can also be used to compare data and check the consistency of interviews across all interviewers and teams. This can help to identify any inconsistencies in the understanding of the questionnaire across teams, or to uncover any cheating by interviewers or teams.

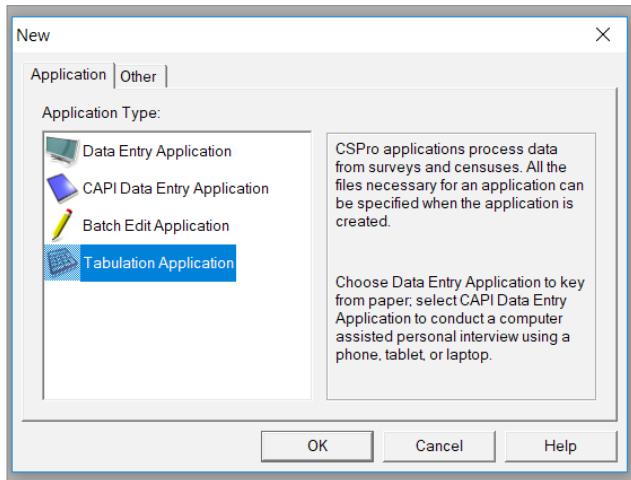
Figure 9.9: Example of Field Notes Displayed in CSPro Text Viewer

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

To set up quality control tables in CSPro, open a new project and click “Tabulation Application” (Figure 9.10). Note that if users are more familiar with tabulations in another software package, such as SPSS, data can be exported in the appropriate format to create tabulations.

After giving a name to the new project, the data dictionary from the original project needs to be selected (Figure 9.11).

Figure 9.10: Creating Quality Control Tables in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Items and variables from the data dictionary can then be dragged from the tree on the left of screen to the table (Figure 9.12). There is also a working storage for the table if any additional variables for cross-tabulation are required.

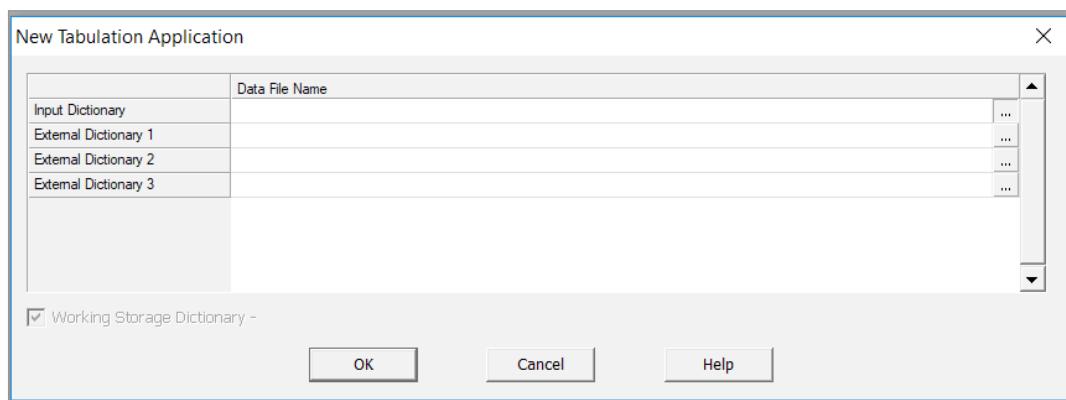
Once the required variables are in place, the tables can be run by selecting the traffic light icon in the icon bar. Be sure to select the most recent data file downloaded from the server. Figure 9.13 shows an example of the table produced once the process is complete.

9.4 Batch Editing for Error Checks

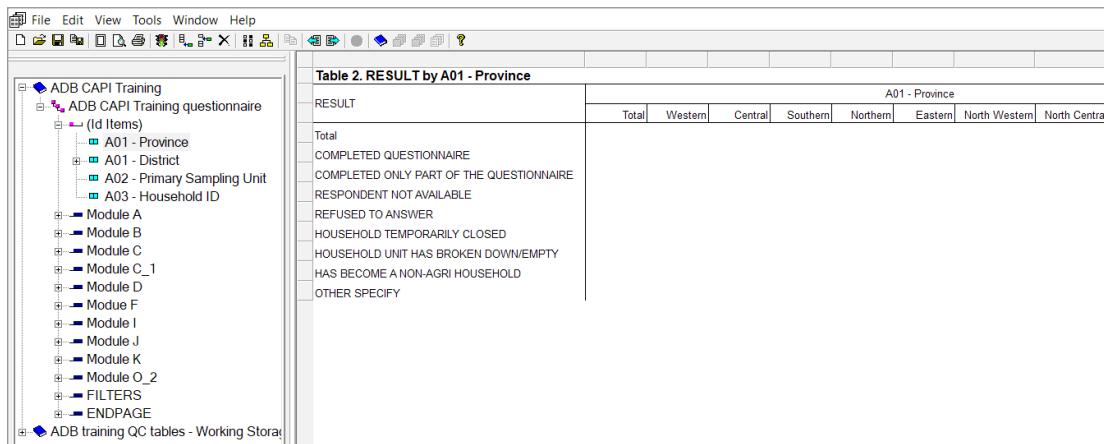
Batch editing functionality in CSPro can be utilized to check through all of the received data for outliers or erroneous answers. A report is then produced by the system, identifying any individual value that has been flagged by the conditions created in the batch editing process.

To create a batch editing process for your survey, start a new CSPro project and press “Batch Edit Application” (Figure 9.14).

Figure 9.11: Selecting a Data Dictionary for Tabulation



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.12: Dragging Variables from the Data Dictionary to the Tables

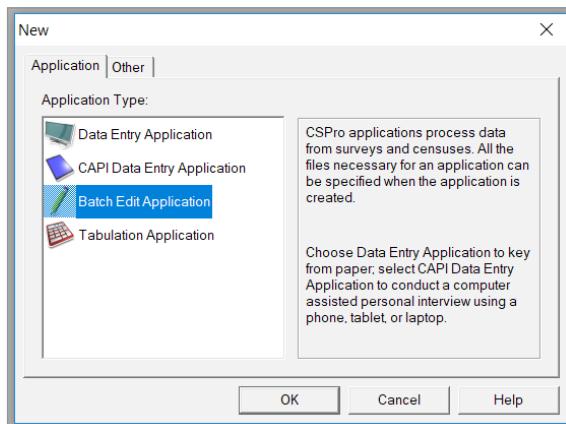
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.13: Example of a Quality Control Table in CSPro

The screenshot shows a quality control table titled "Table 2. RESULT by A01 - Province". The table has a header row with columns for "RESULT" and "A01 - Province". Below this, there are several rows of results with counts for each category across the specified regions.

RESULT	A01 - Province						
	Total	Western	Central	Southern	Northern	Eastern	North Western
Total	3	1	-	1	-	-	1
COMPLETED QUESTIONNAIRE	2	-	-	1	-	-	1
COMPLETED ONLY PART OF THE QUESTIONNAIRE	-	-	-	-	-	-	-
RESPONDENT NOT AVAILABLE	1	1	-	-	-	-	-
REFUSED TO ANSWER	-	-	-	-	-	-	-
HOUSEHOLD TEMPORARILY CLOSED	-	-	-	-	-	-	-
HOUSEHOLD UNIT HAS BROKEN DOWN/EMPTY	-	-	-	-	-	-	-
HAS BECOME A NON-AGRI HOUSEHOLD	-	-	-	-	-	-	-
OTHER SPECIFY	-	-	-	-	-	-	-

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

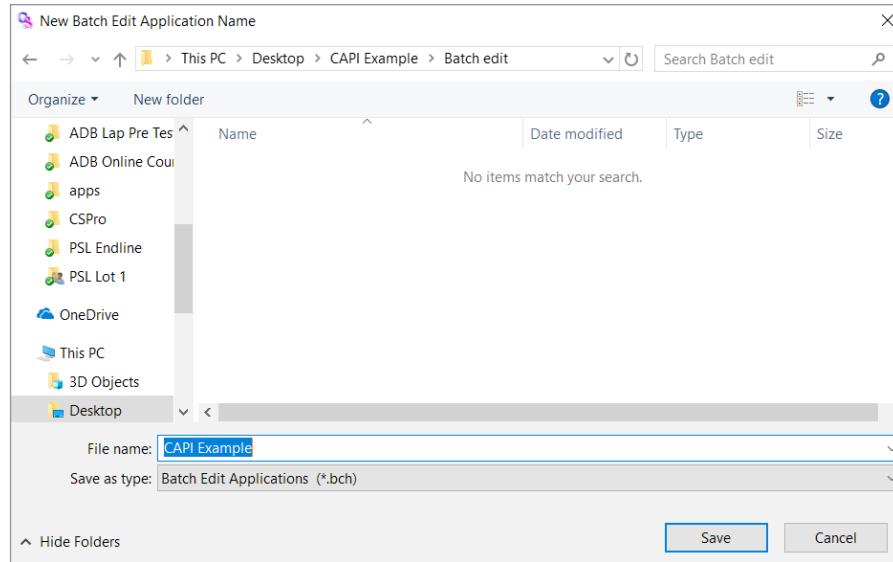
Figure 9.14: Creating a New Batch Editing Process

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

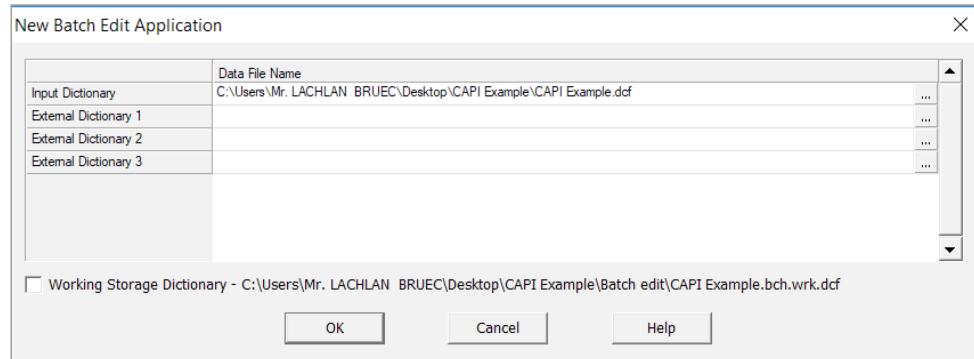
Create a folder and a filename for the new batch edit, then click “Save” (Figure 9.15).

Next, select the data dictionary from the master project (Figure 9.16).

The screen layout is similar to the one for the CAPI project, with the data dictionary tree on the left and space on the right for the logic code (Figure 9.17). Basically, the batch editing process will run the checks created on each case in the downloaded data file.

Figure 9.15: Saving a New Batch Editing Process

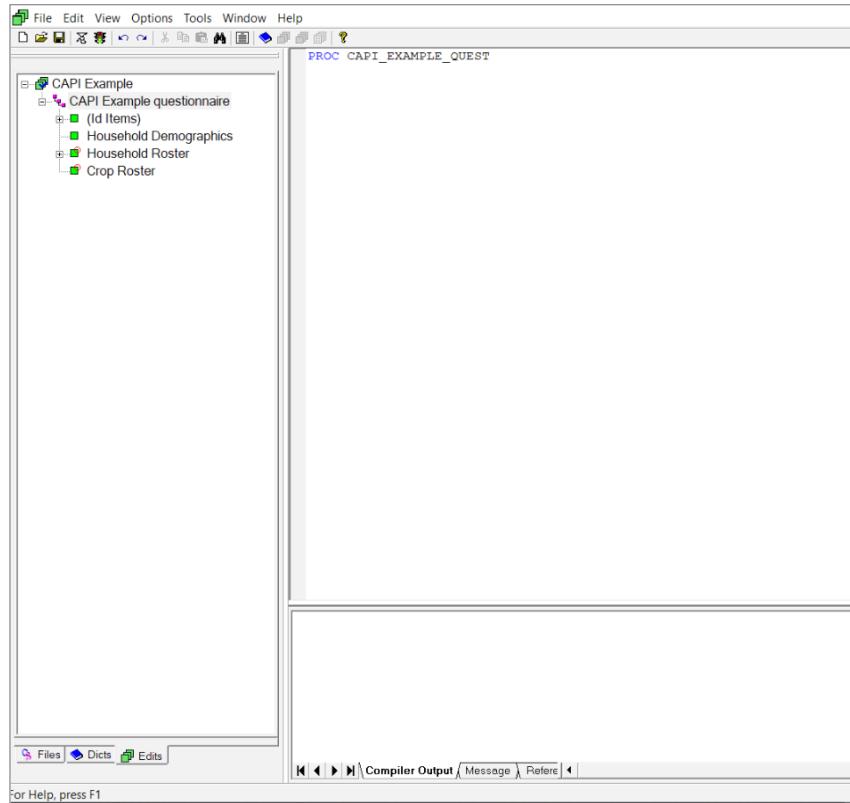
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.16: Specifying the Data Dictionary for Batch Editing

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The logic code is devised in a similar way to how validations are written for the main questionnaire, except there is no need for options such “reenter” or “continue” because the purpose is to flag the case in

a report. An example of logic code for batch editing would be to check for any cases where the household head is under the age of 18 years, if this is not allowed in the survey (Figure 9.18).

Figure 9.17: Screen Layout Used for Batch Editing

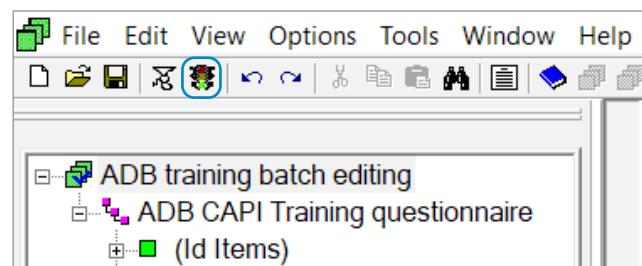
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.18: Example of Logic Code for Batch Editing

```
PROC B06
if B03 = 1 & $ < 18 then
    errmsg ("household head is less than 18 years of age");
endiff
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

The batch editing process can then be run by pressing the traffic light icon in the icon menu (Figure 9.19). If any problem is found, it will be flagged in the report (Figure 9.20).

Figure 9.19: Initiating the Batch Editing Process

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 9.20: Example of Report Generated after Batch Editing

```

CSPro Text Viewer - [ADB training batch editing.lst]
File Edit View Options Window Help
Process Summary
+-----+
| 36 Records Read ( 100% of input file) |
| 0 Ignored ( 0 unknown, 0 erased) |
| 3 Messages ( 3 U, 0 W, 0 E) |
+-----+
Level | Input Case | Bad Struct | Level Post |
+-----+
1 | 3 | 0 | 3 |
+-----+



Process Messages

*** Case [ 3 33 2 2] has 1 messages (0 E / 0 W / 1U)
U -9 Date is earlier than start of fieldwork

*** Case [ 1 13 26 3] has 1 messages (0 E / 0 W / 1U)
U -9 Date is earlier than start of fieldwork

*** Case [ 6 62 3 5] has 1 messages (0 E / 0 W / 1U)
U -15 household head is less than 18 years of age

User unnumbered messages:

Line Freq Pct. Message text Denom
---- ---- -----
9 2 - Date is earlier than start of fieldwork -
15 1 - household head is less than 18 years of age -

CSPro Executor Normal End
-----



For Help, press F1
Sel: (32,1) NUM OVR Encoding: UTF-8 Size: 2653 bytes Pos: (9,1)

```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Such reports can be generated during fieldwork to check if any interviewers or teams are recording a high number of errors. Feedback can then be provided to field staff, so that issues will not continue throughout the duration of the survey.

9.5 Paradata

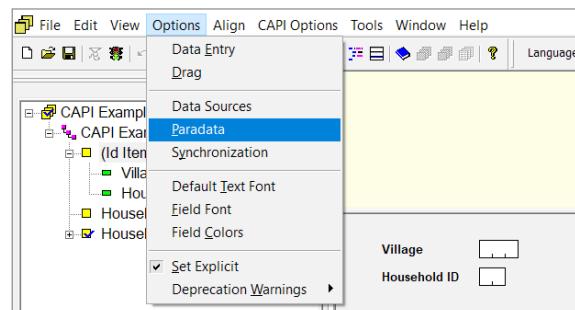
Another quality control feature of CSPro is paradata. Paradata refers to data collected about the interview process itself. It includes information such as the time taken to answer questions, number of errors, tablet versions, information about GPS, and many other aspects of the interviews conducted.

The paradata is captured on each interviewer's tablet and saved in its own file. It can be synced if CSWeb server is being utilized, but will use a lot of bandwidth due to the size of the paradata files. Syncing is therefore not recommended unless the information is critical during fieldwork. Paradata is

often best used after a pretest or pilot test for a survey has been conducted. It can be applied to examine the survey performance for each tablet. In these cases, the paradata can be copied manually from each tablet after returning to the survey headquarters.

First, paradata collection needs to be enabled when designing the master CAPI project. This can be done by selecting “Paradata” from the options menu in CSPro, as shown in Figure 9.21.

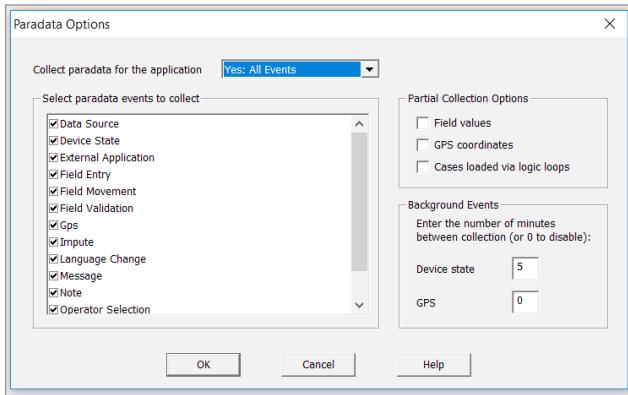
Figure 9.21: Accessing Paradata Options in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

On the next screen, the desired paradata events for collection can be selected by ticking any of the checkboxes, then clicking “OK” (Figure 9.22).

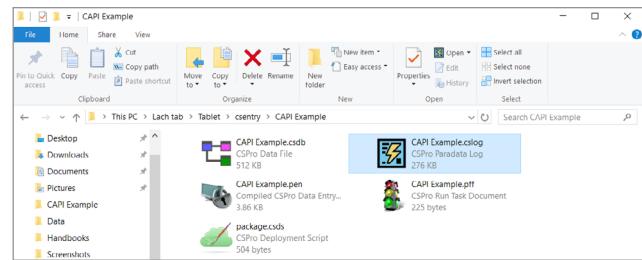
Figure 9.22: Setting the Events for Paradata Collection



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Once some cases have been completed in the field, connect the tablet to a PC using a USB cable, navigate to the “cscopy” folder, then find the folder for the current project (e.g., “CAPI Example”). Inside the project folder, you will find a “.cslog” file (Figure 9.23). Click on this file to open it in CSEntry and view the paradata that has been captured on the tablet connected (Figure 9.24).

Figure 9.23: Accessing the Paradata File on the Tablet



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

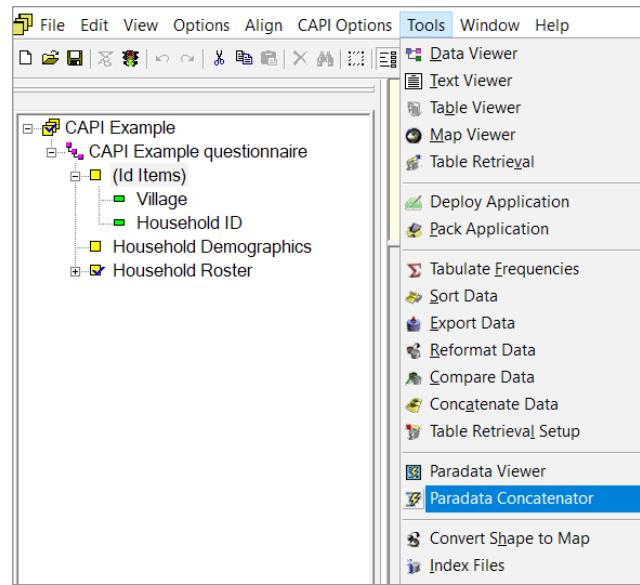
Figure 9.24: Example of Paradata Displayed in CSEntry

Field Name	Number of Fields Entered	Average Duration of Entry (seconds)	Percent of Fields Validated	Percent of Fields Reached via Forward Movement
Q1	11	3.46	100.00%	100.00%
Q2	14	231.59	100.00%	78.57%

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

For some quality control functions, it is more beneficial to view the paradata from many or all of the fieldwork tablets at once, rather than viewing it on a single tablet at a time. If this is required, the “Paradata Concatenator” tool in CSPro can be used to combine paradata files from many tablets into a single file (Figure 9.25).

Figure 9.25: Combining Paradata Files from Multiple Tablets



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 10: Advanced Features

The following are some advanced features of CSPro that may be useful for projects with specific needs.

10.1 Timestamps

A timestamp is a variable that holds the time and/or date of when a specified task takes place. The most common examples are timestamps for the beginning and end points of each interview case.

Collecting timestamps serves two main functions. First, they are used to calculate the duration of each interview. Very short interview durations could be indicative of interviewers rushing through questions, cheating, or creating fake interview cases. Short durations may also be a sign of respondent discontent with the questions being asked, leading to early termination of the interview. The second function of timestamps is to observe the time of day the interviews are completed. For example, if an interview is completed late at night, and the approved practice is for the fieldwork to take place during the day, it may be suspicious and indicative of cheating.

The functions for creating timestamps in CSPro are:

The “timestring” function. This captures the interview time as Unix time, which is the number of seconds that have elapsed since 1 January 1970 (e.g., “1535917968”). This output does not have any meaning when viewed as an integer, but it is very helpful for calculating the difference between two times.

The “sysime” and “sysdate” functions. These capture the current time or date in the format of HHMMSS (hours, minutes, seconds) or DDMMYYYY (day, month, year). They are better used if the time or date of a particular event is of interest, rather than the difference between two times.

To capture timestamps, numeric variables need to be added to the data dictionary, but they need not be included on the forms themselves. This is because the data will be captured in the background and timestamps are not questions for which an answer needs to be selected.

One important thing to note when setting up timestamps is that they reset by default if the interview case is reopened any time after the interview takes place. To prevent this, a simple “if” statement can be used with the value of “notappl”, which means there is no answer input by the interviewer (Figure 10.1). So, if the timestamp variable is empty (i.e., the interview case is opened for the first time), then it will be populated. If it is not empty (i.e., any time after the interview), then it will not be reset.

Figure 10.1: Example of Basic Syntax to Capture a Timestamp

```
if STARTTIME = notappl then
    STARTTIME = timestamp();
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

10.2 Capturing Location Coordinates

Capturing the GPS coordinates of an interview, as well as other location variables such as field perimeters, is one of the key benefits of using CAPI.

To capture GPS coordinates in CSPro, the first step is to create variables to store the GPS data. For simple purposes, latitude and longitude are needed, but more advanced system designers can also collect information about altitude, satellites, accuracy, and read times. The formats for latitude and longitude in the data dictionary are shown in Figure 10.2 and include the specifications for length and number of decimals.

Figure 10.2: Setting Latitude and Longitude Variables

<input type="checkbox"/> LATITUDE	LATITUDE	316	9	Num	Item	1	6	Yes	No
<input type="checkbox"/> LONGITUDE	LONGITUDE	325	10	Num	Item	1	6	Yes	No

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

After the variables are set, a GPS function can be written or copied from a previous project. This function is a piece of code inserted into the Proc Global section of the CSPro logic (Figure 10.3). It can then be called from elsewhere in the system. The “GETGPS” function opens the tablet’s GPS receiver, then allows 60 seconds to read a signal. It is recommended to test this in the field and, if the signal is weak, then the 60 seconds can be increased to allow more time for a better reading. Once a signal is found, the latitude and longitude details are stored in their respective variables. If no signal is found, an error message is displayed to the interviewer, prompting them to try again.

The “GETGPS” function can now be called from a particular question, or from an icon in the menu bar of CSEntry, which can be accessed by the interviewer at any time. Icons in the menu bar are handy for capturing additional survey information, such as GPS coordinates or photos, because a convenient time to do so might vary from interview to interview.

To set up the “userbar” icon in the menu bar, the code in Figure 10.4 can be used in the PreProc of the first form in the questionnaire.

Now, when using CSEntry during interviews, the interviewers will see the userbar icon as highlighted in Figure 10.5.

Figure 10.3: Logic Code to Capture Latitude and Longitude

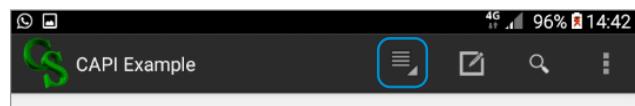
```
PROC GLOBAL
function GETGPS();
gps(open);
if gps(read,60) then
  errmsg("Latitude is %f, longitude is %f",gps(latitude),gps(longitude));
  LATITUDE = gps(latitude);
  LONGITUDE = gps(longitude);
else
  errmsg("GPS signal could not be acquired, please make sure you are outside and location services are turned on");
endif;
gps(close);
end;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 10.4: Basic Syntax to Set Up the Userbar Icon

```
userbar	clear;
userbar	add button, "Collect GPS", GETGPS();
userbar	show;
```

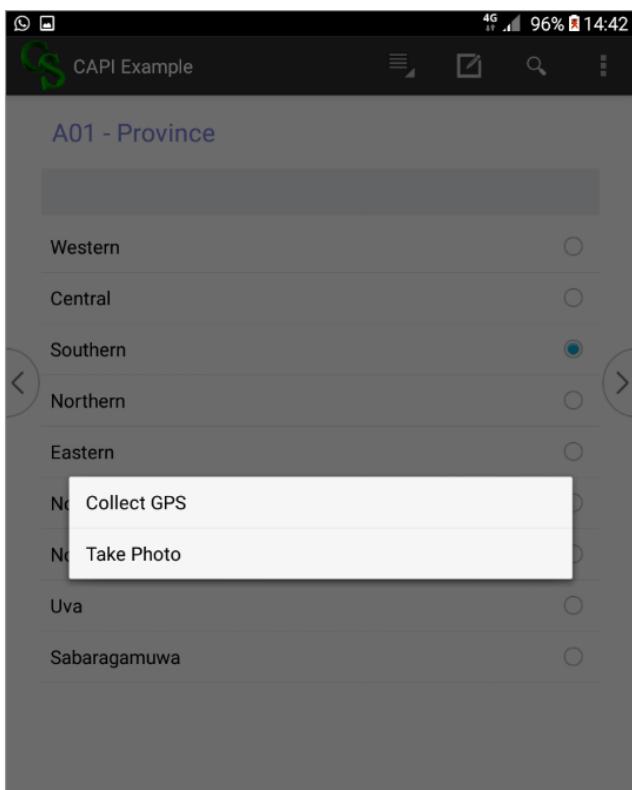
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 10.5: Userbar Icon in CSEntry

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

After pressing the userbar icon, interviewers will see a menu to select any of the user functions that have been added there (Figure 10.6).

Figure 10.6: Accessing the Menu Options in the Userbar



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

10.3 Accessing Other Apps on the Tablet

CSPro can access other apps on the tablet to import additional survey information. The best example of this is accessing the tablet's built-in camera to take photographs of respondents, the household, or other objects of interest. This can be done using the “execsystem” function, and the basic syntax for the code is shown in Figure 10.7.

Figure 10.7: Basic Syntax to Access Other Apps

execsystem(“application”)

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

However, taking a photograph and storing it as part of a survey may not be as simple as first anticipated. If the built-in camera is opened and a picture taken as normal, the image is just stored with all the other photos on the tablet and is given a generic filename. This will then make it very difficult to tell which photo relates to which interview case.

The code in Figure 10.8 (written in the Proc Global section of the CSPro logic) allows the camera to take a photo with a filename that matches with

Figure 10.8: Example of Logic Code for Taking a Photo

```
numeric photocount;

function PHOTO();
string photoFilename = maketext("%sphotos/photo-%d-%02d-%03d-%04d-%05d.jpg",
                                 pathname(Application),
                                 A01, A01_DIS, A02, A03, photocount);
execsystem(maketext("camera:%s", photoFilename));
photocount = photocount + 1
end;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

the questionnaire's identifiers (e.g., "A01" to "A03"). It also includes an incrementing "photocount" function, so that multiple photos can be taken from interview to interview, without overwriting the previous ones.

Similar to capturing GPS coordinates, this function can then be called from anywhere in the questionnaire or accessed from the userbar icon.

10.4 Lookup Tables

A lookup table can be used to search through a large amount of external data for use in validations. In a CAPI system, lookup tables are most commonly used to cross-reference regional identifier codes or to check the prices of different items in a roster.

In the example shown in Figure 10.9, an external lookup file is being used to check that the regional identifiers entered are valid and match with the cluster number.

Figure 10.9: External File Used for a Lookup Table (Excel)

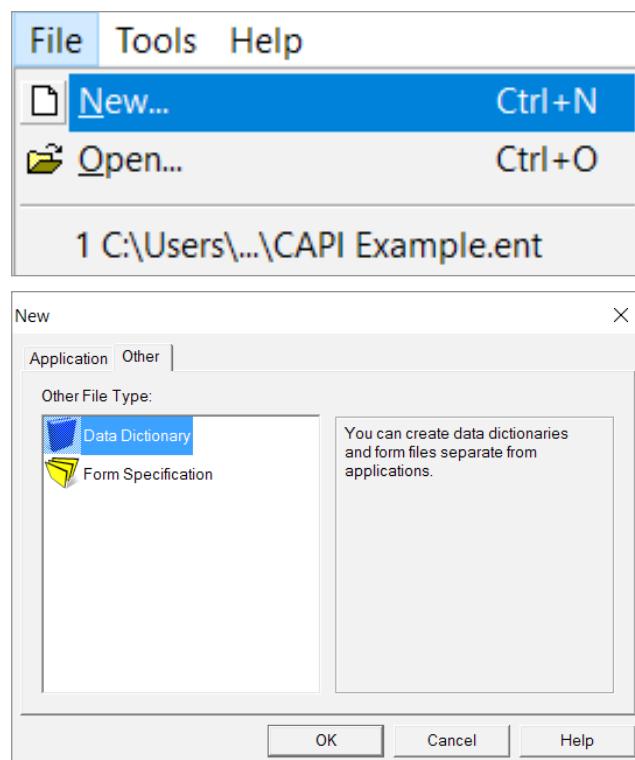
	A	B	C	D	E	F
1	Cluster	State	District	Township	Ward	EA
2	0002	01	01	01	106	001
3	0006	01	01	01	118	010
4	0014	01	01	01	125	023
5	0023	01	01	04	705	002
6	0035	01	02	02	103	005
7	0043	01	02	03	703	004
8	0053	01	03	01	108	003
9	0063	01	03	03	714	003
10	0082	02	01	01	104	003

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

10.4.1 Creating the Data Dictionary for a Lookup Table

To use such a file in CSPro, a new data dictionary needs to be set up to describe the external file. To do this, select "File", then select "New", click on the "Other" tab and select "Data Dictionary" (Figure 10.10). You will be asked to give the new data dictionary a name. Under the example, the dictionary is named "regionlookup."

Figure 10.10: Creating a New Data Dictionary for a Lookup Table



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

You now need to set up variables in the data dictionary to describe the external data (Figure 10.11). The first column where the cluster is stored will be the ID variable. Rename this variable and set the width to "4". Next, set the "record type" (first row) to "0", so that the start of "CLUSTER" is at "1" (the first character in each row of the external file).

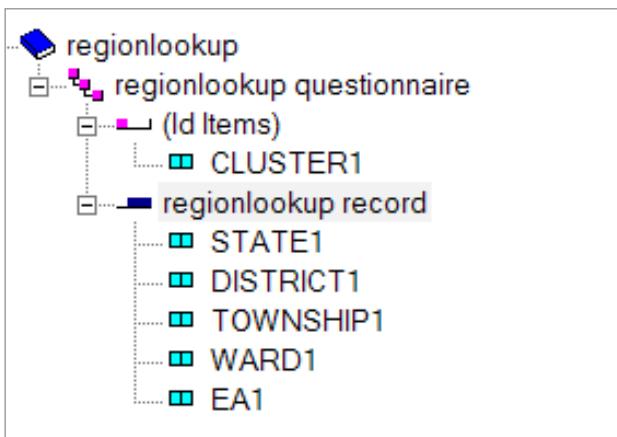
Figure 10.11: Defining the Variables for a Lookup Table

The screenshot shows a data dictionary table with the following columns: Item Label, Item Name, Start, Len, Data Type, Item Type, Occ, Dec, Dec Char, and Zero Fill. There are two rows. The first row is for 'N Item Label (record type)' and the second row is for 'CLUSTER'. The 'CLUSTER' row has values: REGIONLOO, 1, 4, Num, Item, 1, 0, No, and No.

Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
(record type)		0	0	Alpha					
CLUSTER	REGIONLOO	1	4	Num	Item	1	0	No	No

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

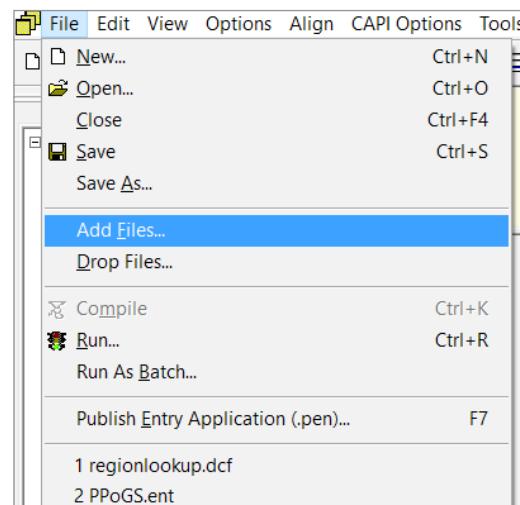
Add the remaining variables with their corresponding correct width. Adding a “1” or some other identifier at the end of each variable will prevent confusion with existing variables later on. When you are finished, your data dictionary will be similar to the example shown in Figure 10.12.

Figure 10.12: Structure of Completed Data Dictionary for a Lookup Table

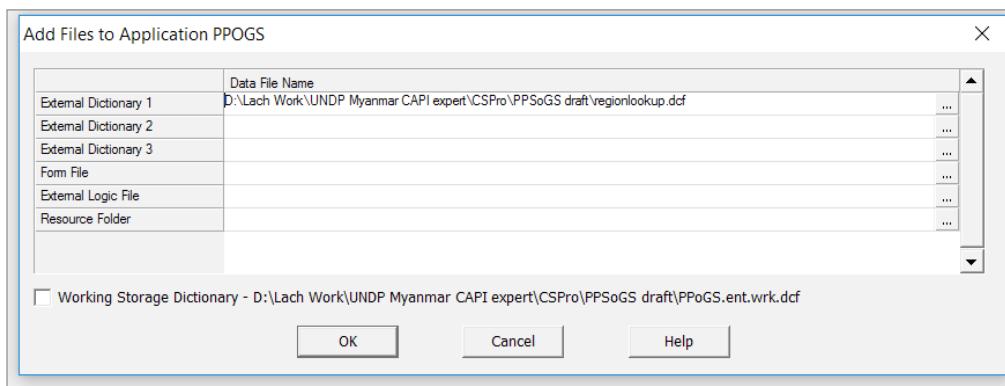
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

After the data dictionary is complete, open the master CAPI project and add the new dictionary by clicking “File”, then “Add Files (Figure 10.13).

You can now specify the new data dictionary (e.g., “regionlookup”) as an external dictionary (Figure 10.14).

Figure 10.13: Adding the Data Dictionary for the Lookup Table

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

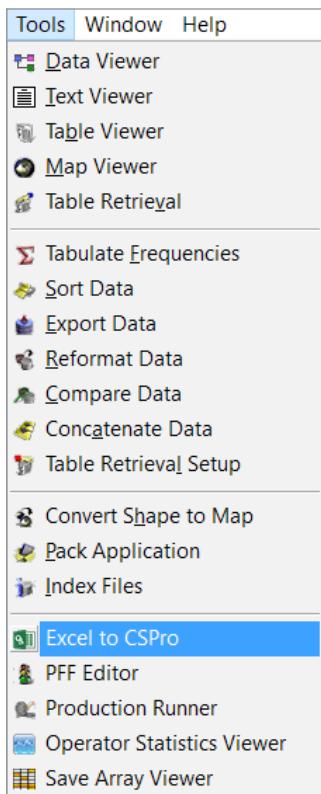
Figure 10.14: Setting an External Dictionary for a Lookup Table

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

10.4.2. Formatting the Lookup File for CSPro

The next step is to convert the Excel lookup file to CSPro format. To do this, use the “Excel to CSPro” tool (Figure 10.15).

Figure 10.15: Converting the Excel Lookup File to CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Now, when you run the lookup table, CSPro will ask you to specify the external file. At this point, select the file just created with the “Excel to CSPro” tool (Figure 10.16).

10.4.3 Writing Logic Code for the Lookup Table

Now that the lookup file is set, it is time to write some logic code that will scan the external data file and return the value. This is done using the “loadcase” function (Figure 10.17).

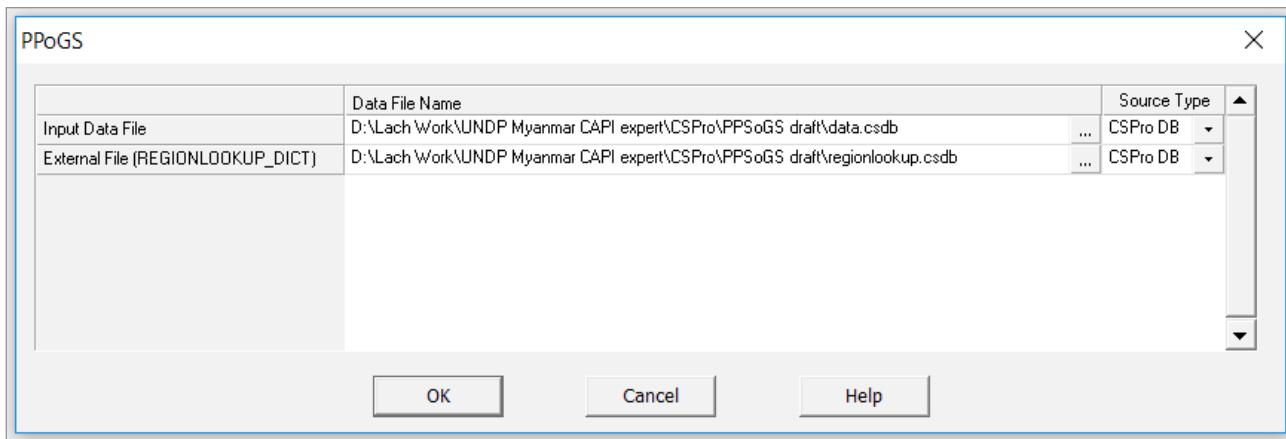
Figure 10.17: Basic Syntax to Scan a Lookup File

loadcase(DICTIONARY, VALUE)

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Figure 10.18 outlines some logic code for the example under discussion. The “loadcase” function will search the external data file and load the row where the cluster matches with the value from “SOQ1” (the cluster entered by the interviewer). Within the code in Figure 10.18 (run in the PostProc section of the question to identify which state the respondent lives in), the “if” statement compares the

Figure 10.16: Selecting the External Data File for Lookup



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

current value (the state entered by the interviewer) to the “STATE1” value from the lookup file. If the wrong state for that cluster ID has been entered, an error message will appear.

Figure 10.18: Example of Logic Code to Utilize a Lookup Table

```
loadcase(REGIONLOOKUP_DICT, S0Q1);
if $ <> STATE1 then
    errmsg ("State does not match cluster");
    reenter;
endif;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

10.5 Task Menus

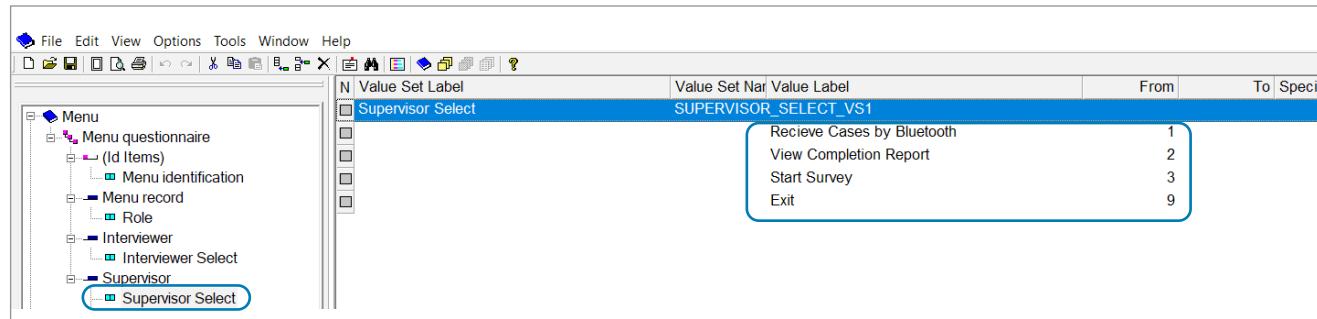
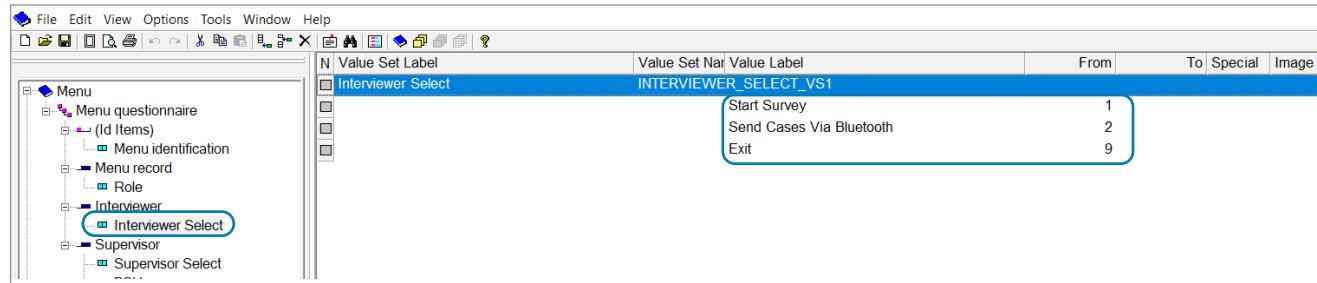
For CAPI projects in which some of the interviewer tasks may be complicated or repetitive, a quick-access menu can be created to launch tasks from a readily available screen. Some of the functionalities of the task menu include:

- Managing sign-ins and passwords for interviewers and/or supervisors
- Creating and viewing reports
- Launching other CSPro tools
- Prefilling ID items (for longitudinal surveys)
- Syncing survey data to the server

To create a task menu, start CSPro and create a new CAPI application. Set the menu options you would like to have in a new data dictionary for each screen in the menu, where the items relate to the tasks you would like to run. Figure 10.19 offers an example of a simple menu with different submenus for interviewers and supervisors.

One of the most common actions from a menu is to start an interview (e.g., “Start Survey”). Basically, when this option is selected, the task menu will close and the “.pff” file for the questionnaire will be launched. The basic syntax to achieve this uses the “execpff” function as shown in Figure 10.20.

Figure 10.19: Example of a Task Menu in CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 10.20: Example of Syntax to Launch a Task from a Menu

```
execpff ("CAPI Example.pff",stop);
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

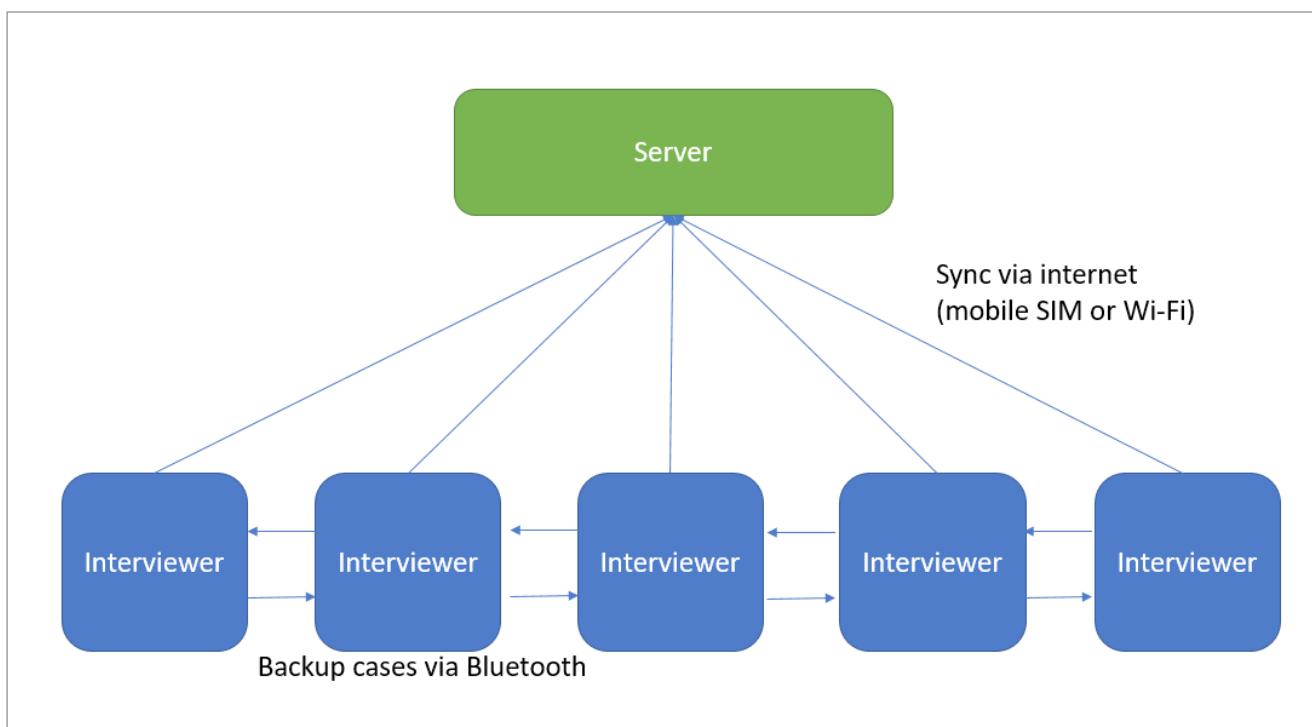
The example in Figure 10.10 (run in the PostProc section of the “Start Survey” menu option) will launch the “.pff” file called “CAPI Example” and also close the menu.

10.6 Bluetooth Syncing

In fieldwork circumstances where there is limited or no internet coverage, Bluetooth syncing can be used to share completed survey data between tablets.

This is beneficial because, if a tablet is lost, stolen, or damaged before internet syncing is possible, there is a risk of losing survey data. Copying the completed interview cases onto multiple tablets reduces this risk. Examples of how Bluetooth syncing can be used in the field are illustrated below.

Figure 10.21 shows a standard syncing setup, where all interviewers can sync their cases directly to the server via the internet, with Bluetooth capability added between each interviewer tablet. Such a setup is useful when internet coverage is generally available, but can be weak or nonexistent at times. The interviewers have the option of using Bluetooth to back up cases to a colleague’s tablet, reducing the risk of lost data, until they return to an area with strong internet coverage, at which point the cases can be synced to the server.

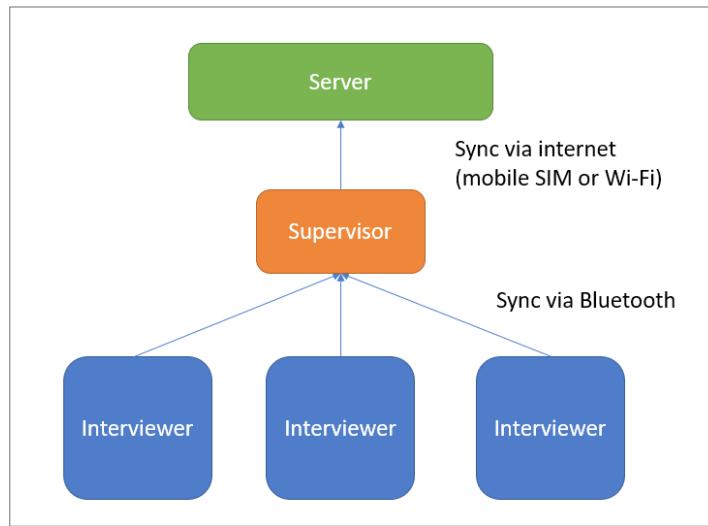
Figure 10.21: Standard Syncing Setup with Bluetooth for Backup

SIM = subscriber identity module.

Source: Graphic constructed by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

In projects for which internet coverage is expected to be weak or nonexistent for the majority of fieldwork, a different syncing setup can be used. Figure 10.22 shows a setup where interviewers can each use Bluetooth to sync their completed cases to the supervisor's tablet. The supervisor can then travel back to an area with internet connectivity and sync all of the transferred cases to the server.

Figure 10.22: Syncing Setup with Bluetooth Connection to Supervisor



SIM = subscriber identity module.

Source: Graphic constructed by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

To achieve Bluetooth syncing, two tablets are needed: one for sending cases (client) and another for receiving them (server). These two tasks will be run using slightly different logic code, so the task menu should contain separate options to send and receive via Bluetooth. The two different sets of logic code (one for sending, one for receiving) are shown in Figure 10.23 and can each be run in the PostProc section of the menu item.

In the field, the tablet users need to be close to one another, have Bluetooth turned on, and run

the send or receive task at the same time. For the recipient, the Bluetooth connection will remain open. For the sender, a pop-up window will appear, asking which device should be connected to which recipient (Figure 10.24). Once the recipient's device is selected, the cases will be automatically synced.

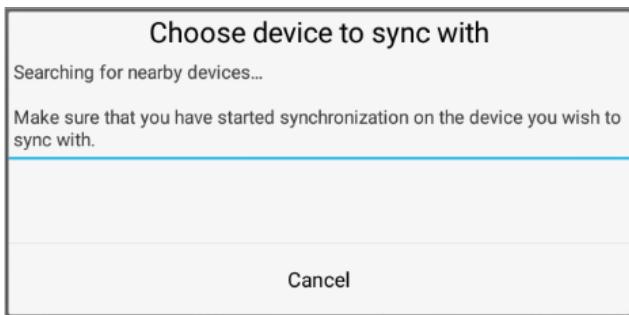
Figure 10.23: Example of Logic Code for Bluetooth Syncing

- Client (sending the cases)


```
if syncconnect(Bluetooth) then;
    syncdata(PUT, DICTIONARY NAME);
    syncdisconnect();
endif;
```
- Server (receiving the cases)


```
syncserver(Bluetooth)
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 10.24: Example of Bluetooth Syncing on CSEntry

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

10.7 Reports

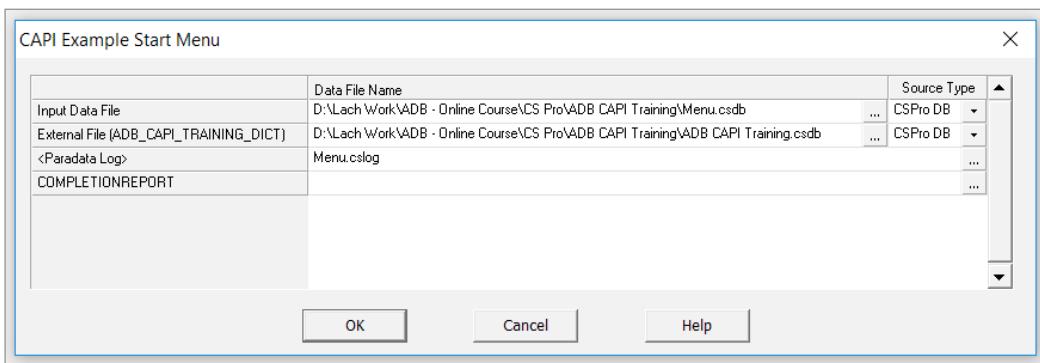
A useful CSPro function that can be set to run from a task menu is to allow supervisors to view progress reports. These reports may summarize the number of completed interview cases or provide other vital information from team members in the field.

Such reports can be generated by using a text file on the tablet and inserting the relevant results.

Figure 10.25: Basic Syntax to Open a Report

```
function REPORT();
open(COMPLETIONREPORT);
```

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

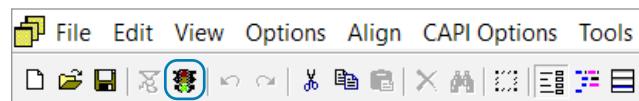
Figure 10.27: Selecting the External Text File to Generate a Report

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

In the example for this section, a completion report is created for a primary sampling unit (PSU). This would be useful for a supervisor to view and ensure that all cases have been captured for a particular PSU, before moving on to the next one.

The first step is to write some function code in the Proc Global section of the logic, then define the external file (blank “.txt” file) that is to be used to print the completion report and display to the user (Figure 10.25). In this instance, the “open” function is used, along with a name to reference the report file in future commands (e.g., “COMPLETIONREPORT”).

Now, run the task on the PC by pressing the traffic light icon (Figure 10.26). The system will ask for the file that will be accessed to generate “COMPLETIONREPORT” (Figure 10.27). Create a new text document (with any name) in the same folder as the other files for the project, then select this file when asked.

Figure 10.26: Running the Task to Generate a Report

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Now that the text file has been created, the next step is to create some text within the report. The “filewrite” command can be used to insert the text (pink text in Figure 10.28) in the “COMPLETIONREPORT” file. The commands in Figure 10.28 establish a heading, a blank line for spacing, and the PSU selected for the report.

Figure 10.28: Example of Logic Code to Insert Text into a Report

```
filewrite (COMPLETIONREPORT, "Progress Report");
filewrite (COMPLETIONREPORT, "");
filewrite (COMPLETIONREPORT, "PSU:%d", PSU);
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Some variables can now be defined to make counts of all the items we would like to display in the report. Figure 10.29 shows the code used to display how many interview cases have been completed, partly completed, respondent not available, or refused.

Figure 10.29: Example of Logic Code to Define Numeric Variables for a Report

```
numeric complete = 0;
numeric part_complete = 0;
numeric responavail = 0;
numeric refused = 0;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

With the variables to hold the count defined and set to “0”, a loop will be created to count each variable for all the interview cases on the tablet running the report. The “forcase” function is used to check each case and incrementally count the variable for each respective state at the variable “RESULT” (Figure 10.30). The loop will run until all cases have been checked.

Figure 10.30: Logic Code to Count Variable Frequencies

```
forcase CAPI_EXAMPLE_DICT do
    if RESULT = 1 and A02 = PSU then
        complete = complete + 1;
    elseif RESULT = 2 and A02 = PSU then
        part_complete = part_complete + 1
    elseif RESULT = 3 and A02 = PSU then
        responavail = responavail + 1
    elseif RESULT = 4 and A02 = PSU then
        refused = refused + 1
    endif;
endfor;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Now that the count variables have been completed, they can be inserted into the report by using the “filewrite” command and substitution of the count variables (Figure 10.31).

Figure 10.31: Logic Code to Insert Counts into a Report

```
filewrite (COMPLETIONREPORT, "Completes: %d", complete);
filewrite (COMPLETIONREPORT, "Partially Completed: %d", part_complete);
filewrite (COMPLETIONREPORT, "Respondent Not available: %d", responavail);
filewrite (COMPLETIONREPORT, "Refused to answer: %d", refused);
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Lastly, some code must be written to close the report file and open the text file on the tablet using the “execsystem” command (Figure 10.32). The directory needs to be the place where the text file is stored on the tablet. The “end” command tells CSPro that this function is complete.

Figure 10.32: Example of Logic Code to View a Report

```
close (COMPLETIONREPORT);
execsystem("view/storage/emulated/0/csentry/CAPI EXAMPLE/COMPREPORT.txt");
end;
```

Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Now, when the reporting function is called from the task menu, the report will count the cases and display to the user (Figure 10.33).

Figure 10.33: Example of a Report Displayed on a Tablet



Source: Screen shot generated by Asian Development Bank consultant. 2018.
Phnom Penh, Cambodia.

Chapter 11: Help and Support

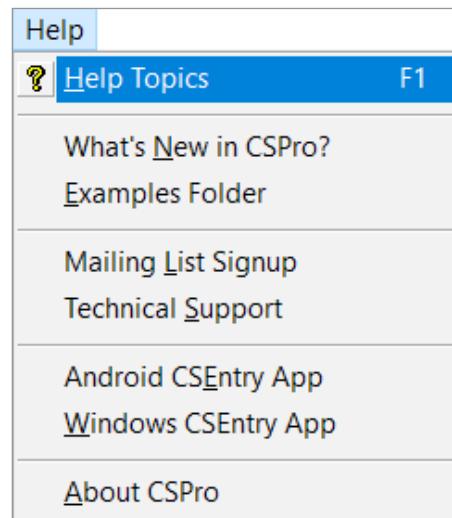
CSPro provides various resources to help and support its users. Built-in and online guides are available to help users solve problems when working on projects and/or to learn more about the advanced features of CSPro. These resources should be utilized before contacting CSPro directly for technical support.

11.1 Help Documentation

CSPro offers excellent help documentation, both contained within the software and available on the web. To access the help documentation within CSPro, go to the help menu and click on “Help Topics” (Figure 11.1).

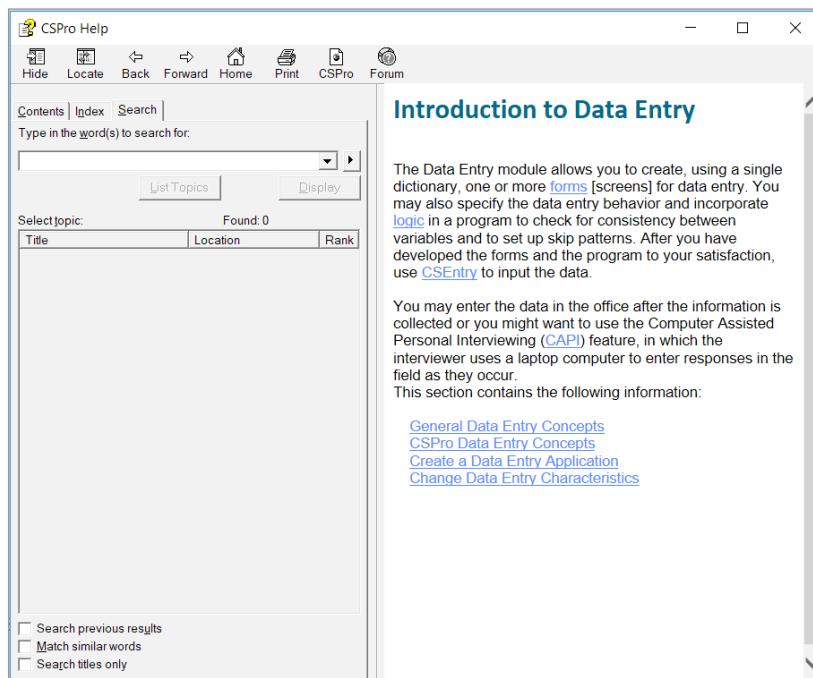
The help section is organized by contents and index, with specific topics accessible by using the search tab (Figure 11.2). Using the search function, users are able to type in any issue or task that may be causing them problems, and find detailed descriptions often accompanied by examples.

Figure 11.1: Accessing Help Topics within CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 11.2: Help Functions and Documentation within CSPro



Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

11.2 Online Users Forum

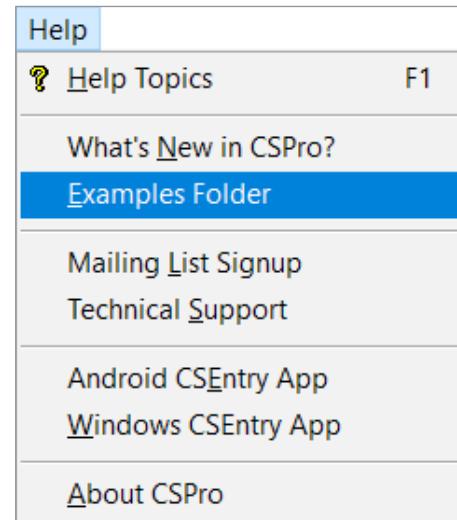
The CSPro Users Forum is a website where users from all around the world can ask and answer questions about the system (Figure 11.3). If you have an issue that you are unable to resolve, the forum allows you to receive feedback and advice from other users or the CSPro support team. You can also scan existing posts to see if someone else faced a similar problem. The forum can be accessed at <http://www.csprusers.org/forum/>.

11.3 Example Projects

Each release of CSPro comes with example projects created by the CSPro development team. These are useful as templates for creating a CAPI project with certain functionalities. To access these, select “Examples Folder” from the help menu (Figure 11.4).

Each project is contained in its own folder and can be opened and run from this directory (Figure 11.5).

Figure 11.4: Accessing Project Examples Created by CSPro



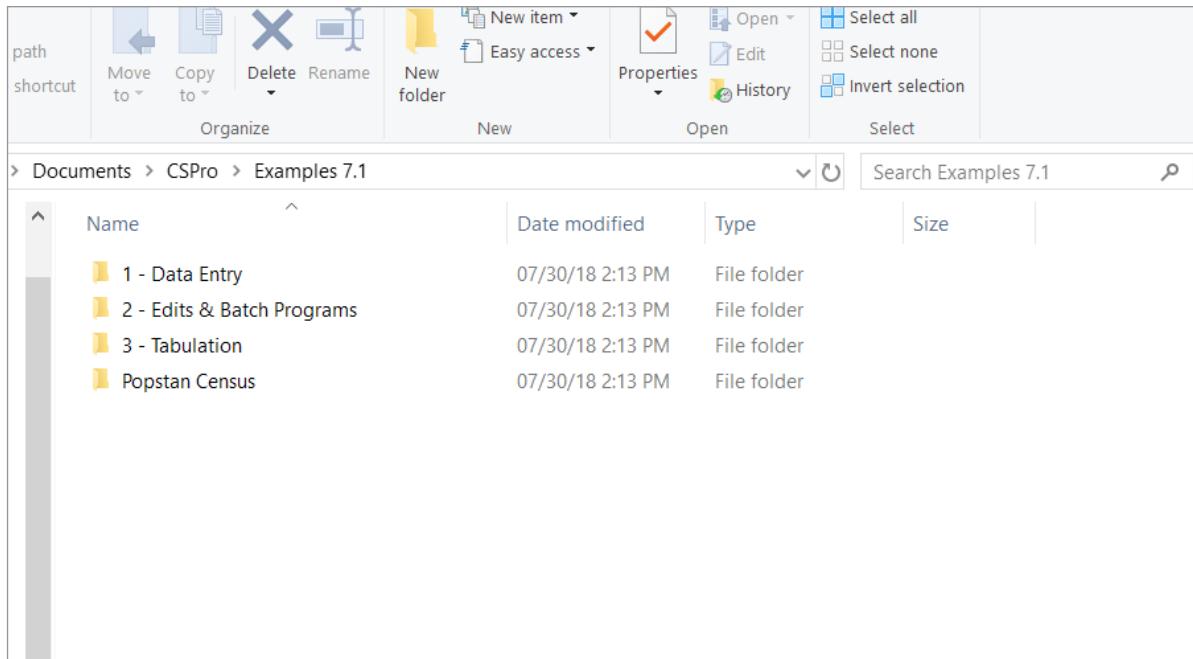
Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 11.3: CSPro Users Forum

A screenshot of the CSPro Users Forum homepage. The header includes the URL 'Not secure | www.csprusers.org/forum/' and a search bar. The main content area shows a list of discussion boards:

FORUM	TOPICS	POSTS	LAST POST
Entry Discussions about CSEntry	1118	4807	Re: A subset of a Multiple re... by Imoriba 11 November 21st, 2018, 4:31 pm
Android Discussions about creating CAPI applications to run on Android devices	158	655	Re: Modify auto Increment by Bhupender11 1 November 23rd, 2018, 1:15 pm
Synchronization Discussions about syncing data via Bluetooth, Dropbox, FTP, and using CSWeb	169	870	Re: Datalviewer doesn't read d... by Jaycan 1 November 15th, 2018, 9:37 am
Editing Discussions about editing and cleaning data	82	303	Re: Batch file to Create new ... by Bruganda 1 November 15th, 2018, 12:25 am
Tabulation Discussions about tabulating data in the designer or in batch mode	61	204	Re: Tabulate frequency by Mariovalisman 1 August 6th, 2018, 1:18 pm
Tools Discussions about tools to complement CSPro data processing	91	347	Re: Excel to CSPro produces e... by Gregory Martin 1 September 26th, 2018, 7:35 am
Other Other discussions about CSPro	194	770	Re: Multi select Options at t... by josh 1 November 23rd, 2018, 7:17 am
Feature Requests What would you like to see in CSPro?	123	402	Re: Improve Auto-increment by Bhupender11 1 November 22nd, 2018, 6:51 am
News CSPro News	24	49	Retrieving overwritten data i... by Mase 1 July 18th, 2018, 6:36 am

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

Figure 11.5: Opening Project Examples from the Systems Directory

Source: Screen shot generated by Asian Development Bank consultant. 2018. Phnom Penh, Cambodia.

References

- Asian Development Bank (ADB). 2018. Graphic Constructed by ADB Consultant. Phnom Penh, Cambodia.
- _____. 2018. Screen Shot Generated by ADB Consultant. Phnom Penh, Cambodia.
- _____. 2018. Table Constructed by ADB Consultant. Phnom Penh, Cambodia.
- _____. 2019. Screen Shot Generated by ADB Consultant. Newcastle, United Kingdom.
- United States Census Bureau (USCB). Census and Survey Processing System. <https://www.census.gov/data/software/cspro.html>.

Conducting tablet-based field data collection with CSPro

A Handbook

Conducting tablet-based field data collection with CSPro: A Handbook is a joint initiative of the Asian Development Bank and the Food and Agriculture Organization of the United Nations to support national statistics offices and line ministries to develop human capacities to conduct tablet-based field data collections for official statistics in the Asia and Pacific region for more robust, accurate and timely data.

The adoption of tablet-based data collection methods, also referred to as Computer-Assisted Personal Interviewing (CAPI), is part of an overarching development in official statistics to adopt new cost-effective technologies to move from traditional pen and paper questionnaires to more cost-efficient, high quality and timely methods using electronic devices.

This Handbook seeks to support this transition by providing step-by-step instruction and guidance to develop, test and run CAPI field data collection using one of the free software's currently available on the market – CSPro.

About the Asian Development Bank

ADB is committed to achieving a prosperous, inclusive, resilient, and sustainable Asia and the Pacific, while sustaining its efforts to eradicate extreme poverty. Established in 1966, it is owned by 68 members—49 from the region. Its main instruments for helping its developing member countries are policy dialogue, loans, equity investments, guarantees, grants, and technical assistance.

About the Food and Agriculture Organization of the United Nations

FAO is a specialized agency of the United Nations that leads international efforts to defeat hunger.

Our goal is to achieve food security for all and make sure that people have regular access to enough high-quality food to lead active, healthy lives. With over 194 member states, FAO works in over 130 countries worldwide. We believe that everyone can play a part in ending hunger.

FAO REGIONAL OFFICE FOR ASIA AND THE PACIFIC

39 Phra Athit Road, Phra Nakhon
Bangkok 10200
www.fao.org/asiapacific

ISBN 978-92-5-132179-9



9 789251 321799

CA7690EN/1/05.20

ASIAN DEVELOPMENT BANK

6 ADB Avenue, Mandaluyong City
1550 Metro Manila, Philippines
www.adb.org