

Session 1: Overview of CSPro, Dictionary and Forms

At the end of this lesson participants will be able to:

- Identify different CSPro modules and tools and their roles in the survey workflow
- Create a simple data entry application including dictionary and forms
- Run a data entry application on Windows
- Run a data entry application on Android and retrieve the data entered
- View data using DataViewer
- Understand the differences between the new CSPro DB format and the old text format for data files

CSPro Overview

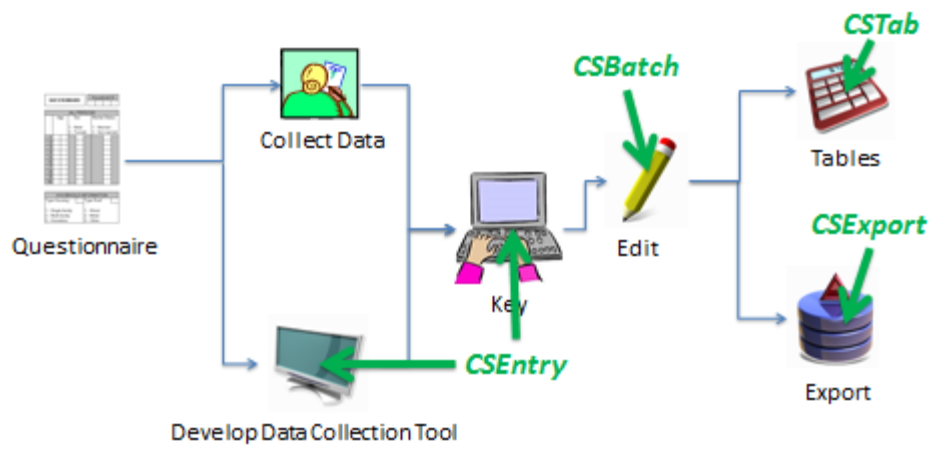
CSPro is a suite of software tools for census and survey data processing that includes modules for data collection, editing, tabulation, and dissemination.

CSPro has a long history. It was first released in 2000 and has been used in over 100 countries worldwide. It has been used for censuses all over the world as well as for many large and complex household surveys including the Demographic and Health Survey (USAID), Multiple Indicator Cluster Survey (UNICEF) and Living Standards Measurement Study (World Bank). The first Android version was released in 2014. CSPro Android has already been used in production for household surveys and population censuses in multiple countries.

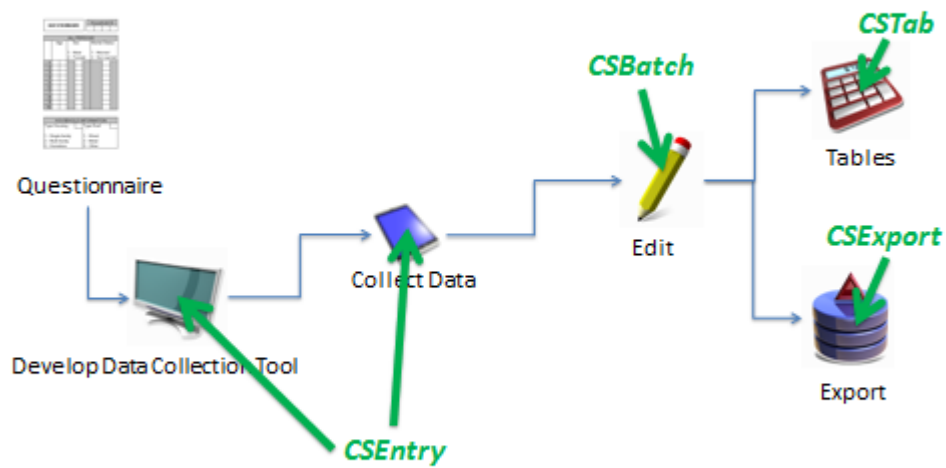
CSPro is free software developed by the US Census Bureau and funded by USAID. The Census Bureau provides free email customer support. You can send questions to cspro@lists.census.gov.

CSPro can be used for both the traditional PAPI (pencil and paper interview) workflow as well as the computer aided personal interview (CAPI) workflow. In this workshop, we will focus on data collection in using a CAPI workflow.

Paper and Pencil Workflow



CAPI Workflow



Using CSPro Android

Before we learn how to build data entry applications in CSPro, let's try doing some data collection to get comfortable with the system.

Group exercise

Split into groups of 3-4 people and use the provided tablets to interview each other using the "Getting to Know You" application. Interview each member of the group so that we have data for all workshop participants. When you are done, tap the sync button (🔄) to upload your results to the server.

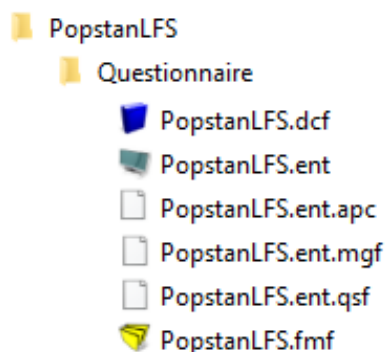
Creating a Data Entry Application

For most of the examples in this workshop, we will be creating a data entry application based on the labor force survey questionnaire included with the workshop materials. With CAPI applications, a paper questionnaire alone is not sufficient to define a data entry application. We also need a specification document that describes consistency checks, skip patterns, text fills, error messages and other aspects of the interactive application that are not defined on a paper questionnaire. Take a moment to review both the questionnaire modules and the accompanying specification documents. For our examples, we will use the fictitious country Popstan for our survey.

When you launch CSPro you are given the choice of "Data Entry Application" for key from paper (PAPI) and "CAPI Data Entry Application" for electronic data collection using phones/tablets/laptops. The differences are:

- System controlled (tightly controlled path)
- CAPI question text
- Extended controls (radio buttons, checkboxes, date picker, ...)

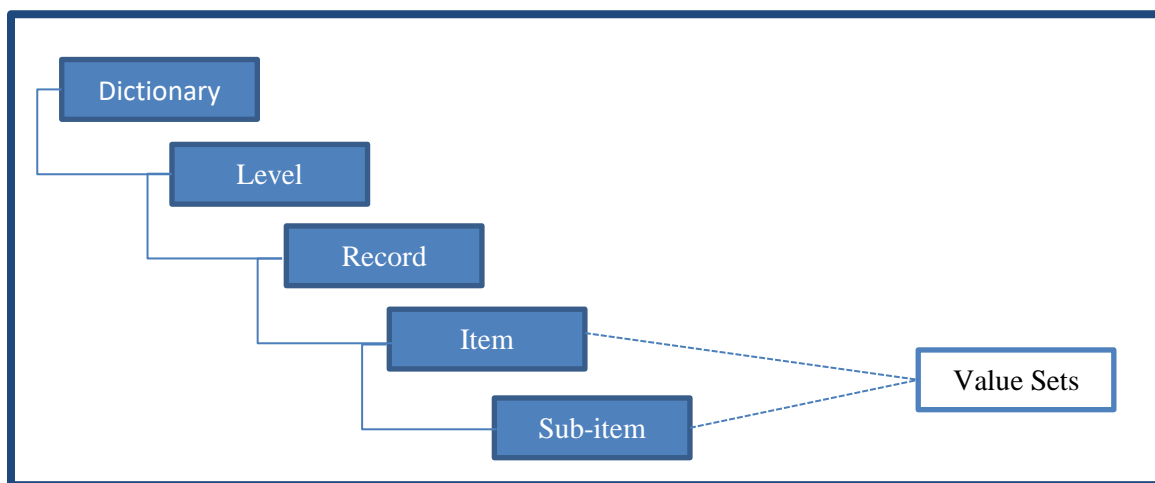
Since we are creating a CAPI application we will choose "CAPI Data Entry Application". We will name the application "PopstanLFS" and we will use the same name for the dictionary. Since we will eventually add other applications such as the menu, we will create the following folder structure:



Tip: CPro generates lots of files when you create and run applications and uses the file extensions (.apc, .ent,, .mgf...) to distinguish. It helps to enable showing file extensions in Windows so that you can tell which file is which.

The Data Dictionary

The first step in creating the application is to define the data dictionary. The data dictionary lists all the data items and possible responses that will be in the application and organizes them in records and levels. The dictionary has the following hierarchy:



Before defining the record and items we first create **ID-items**. **ID-items** uniquely define each case (each questionnaire). Usually these are geographic codes.

What are the id-items for the example questionnaire? The geographic and household identifiers section contains many variables but to uniquely identify a questionnaire we only require the geographic identifiers (province, district), the dwelling unit number and the household number. The rest of these values do not help distinguish one household from another in the database of households. They will eventually be filled in automatically by the case management system.

Add the id-items to the dictionary

- Province (numeric, length 1)
- District (numeric, length 2)
- Dwelling unit number (numeric, length 3)
- Household number (numeric, length 3)

Properties of dictionary items:

- Label: text displayed to interviewers (not the full question text, we will see where to put that in a later lesson).
- Name: used to refer to variable in CSPro logic (interviewers will never see this)
- Start: column number of first digit (character) of variable
- Len: number of characters/digits used to store variable
- Data type: alpha for names and other text, numeric for numbers and coded responses
- Item type: whether variable is item or sub-item
- Occ: number of occurrences for variables that are repeated in record
- Dec: for numbers with fractional parts, number of digits after decimal point
- Dec char: whether or not to include decimal point in decimal numbers
- Zero fill: for numeric values whether to add zeros or blanks to left of number when number of digits in value is less than the length of the variable. Using this will avoid problems when dealing with subitems. We can enable zero fill by default on the options menu.

Tip: Note that you can toggle showing names or labels in the dictionary tree on the left side of the screen using the View menu. You can also select "Append names to labels in tree" to show both at the same time.

What are the records used in our survey?

We could use a separate record for each section of the questionnaire (demographics, proxy, employment, absence...). We could also use a single record for nearly all the sections in the questionnaire that have to do with the individual. You generally only need to put variables on separate records if you expect to have different numbers of occurrences for different types of records. For example, in a household you might require up to 20 occurrences of the record for information about individuals but you would only have one record occurrence for questions regarding the characteristics of the housing unit.

In our case we will need one record for the additional geographic and household identifiers, one for the demographic section (household roster) and another for the remaining sections regarding the individual. We could combine the demographics sections with the other individual questions but as we will see later on this would make it difficult to be able to ask the demographic questions and the other questions in different orders. To allow us to always start the demographics section with the head of household but potentially start the proxy section with the person who completed the household roster, we need to have two separate records.

This leaves us with the following three records

- Household and Geographic Identifiers
- Demographics
- Labor

Let's start by just creating the household and Geographic identifiers and Demographics records.

Properties of records:

- Label/name: same as for variables.
- Type Value: to distinguish between different records in the data file.
- Required: whether or not the record must be entered for the questionnaire to be complete.
- Max: for multiply occurring records the maximum number of occurrences allowed. Generally, 1 for singly occurring records (like housing) and a larger number for repeating records (like 20 for household members).

What are the properties of the Geographic and Household Identifiers record?

- Name: GH_REC
- Type value: G (can use anything but nice to use something meaningful like first letter of name)
- Required: yes
- Max: 1

And the demographics record?

- Name: DEM_REC
- Type value: D
- Required: yes (unless we can have empty households)
- Max: 20

Add some fields to the demographics record:

- PPNO Person number (numeric length 2)¹
- NAME Name (alpha length 30)
- DEMSEX Sex (numeric length 1)
- DEMREL Relationship (numeric length 1)
- DEMAG1 Age (numeric length 3)

And to the geographic and household identifiers record:

- GHAREA Type of Area (numeric length 1)
- GHVFDT Visit date (numeric length 8)

A note on variable naming

Different people have different styles of naming dictionary variables. Some use a descriptive name such as "PLACE_OF_BIRTH" others prefer to use the question number such as "B07" and others prefer a

¹ The line number is not needed in CSPro itself as there are ways to determine the row number using logic however when exporting the data to other packages it is often useful to have it. We will see later how to fill this in automatically during data entry.

combination such as “B07_PLACE_OF_BIRTH”. Whichever approach you choose just make sure that it will be easy for users of your application and your data to understand. Will everyone working on the logic for your application know what B07 is?

For each of our variables we need to add the possible responses (value sets). The value set lists all valid responses along with their corresponding labels for coded variables. Without a value set, the interviewer can enter any value (except blank) but with a value set they are limited to the options defined in the value set. Without a value set, users can even enter negative numbers. For this reason, it is good practice to use a value set for all numeric variables.

Define the value sets for some of our variables based on the response codes on the questionnaire:

- Province (copy/paste from Excel)
- Person number (range: 1-20)
- Name (no value set)
- Relationship (see questionnaire)
- Sex (1- Male, 2-Female)
- Age (use a range: 0-120)

Forms

Before we can enter data, we need to create data entry forms. To start, click on the yellow stack of forms on the toolbar.

Create a form for household and geographic identifiers and drag and drop the id-items onto the form. Note that we can drag and drop individual items or entire records. By right clicking on the form in the forms tree on left side of the screen we can change the label and name of the form. Let’s make the label “Geographic and Household Identifiers” and make the name “GH_FORM”.

Create a form for the household roster and drop the items from the demographics record. Let’s give the form label “Demographics” and name “DEM_FORM”. Note that when we drop the record we have the option to put the items in a roster or a repeating form. If we drop the items on the geographic and household identification form, we can only roster since the household identification isn’t repeated. For our example let’s use a roster.

When we create the rosters, CSPro automatically gives them a name that ends in “000”, for example “DEMOGRAPHICS000”. You can see this in the forms tree on the left side of the screen. We can change this by right clicking on the roster in the forms tree and choosing properties. Let’s name our roster “DEM_ROSTER”.

For paper and pencil surveys, we would spend a lot of time on the layout, adding additional labels and frames to make the form look exactly like the questionnaire. However, when rendered on Android, the form is rendered one question at a time so making the form look like the paper form is not as important.

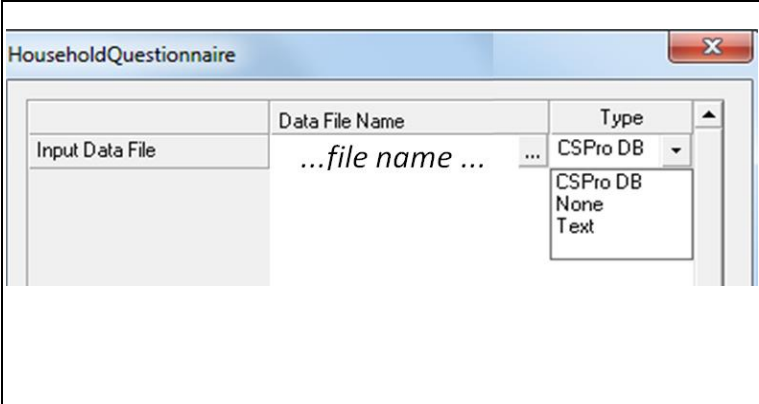
Running the Application on Windows

Click on the green light icon on the toolbar to run the data entry application on your PC. Enter the name of the data file. For data files CSPro uses the file extension “.csdb”. Let’s name our data file “PopstanLFS.csdb”.

Data File Types

The csdb extension is new in CSPro 7.0 and represents a new file format, the CSPro Data Base File. This file not only contains the data itself but it also contains the notes, the index, the partial save status and metadata used for data synchronization. In earlier versions of CSPro, the notes, index and partial save were stored in extra files that accompanied the data file itself. It was unwieldy to deal with all of these files so the CSPro DB file combines them all into one file. Unlike the text file used by earlier versions of CSPro, this is a binary file that cannot be viewed using TextViewer or Notepad. Instead you can view them in the DataView tool.

In CSPro 7.0, when you launch a CSPro data entry application you can select the type of data file that you want to use:

	<p>CSPro DB: This is a CSPro Data Base File. It contains the data and metadata</p> <p>None: There is no associated data file. This is useful when building menus.</p> <p>Text: This is a UTF8 Text file. This is what CSPro has used in previous versions.</p>
--	---

While it is still possible to use text files in CSPro 7.0, it is highly recommended to use CSPro DB instead. There are new features such as smart sync and case labels that are not supported using text files.

Controlling the size of the roster

Our program at this point has no way to stop entering household members unless we fill in all 20 rows of the roster. To limit the number of rows in the roster to the number of actual household members we can have the interviewer enter the number of household members and use that to set the size of the roster. Add a new variable, **number of household members (DEMNMEM)**, to control the size of the roster. Which record should we add it to? It can’t be on the demographic record since we don’t want it to repeat so let’s add it to the household identifiers record but put it on the demographics form. Right click on the roster, choose properties and set this new field as the occurrence control field.

Try running the application again. Unfortunately, when we get to the person form we go into the roster instead of the “number of household members” field. By default, CSEntry goes in the order in which the fields were dropped on the form. We can change the order by dragging the fields in the forms tree. Move the “number of household members” field up in the tree so that it comes before the roster. Now the roster control field should work. In the next lesson, we will use logic to exit the roster without the roster control field.

DataViewer

Once you have run the application and completed a case, you can view the data that was entered by double-clicking on the csdb file to open it in DataViewer.

Running the Application on Android

To put the application on an Android phone/tablet we need to do the following:

1. Publish the .pen file (File menu → Publish Entry Application).
2. Connect the tablet to the PC using a USB cable. The tablet should show up like a USB drive.
3. Copy the files PopstanLFS.pen and PopstanLFS.pff into the CSEntry directory on the Android device.
4. Start CSEntry on the Android device
5. Enter the data (note the differences between Android and Windows)
6. When you are done, connect the tablet back to the computer and copy the file PopstanLFS.csdb from the tablet back to the PC. On some tablets (Nexus 7 for example) the first time the data file is created it may not show up when connected to the PC. If this is the case, reboot the tablet and it will.

When we copied the application to the tablet we copied the pen and the pff file. The pff file contains various parameters about how to launch the data entry application including the data file to use. You can modify the pff file by right clicking on it in Windows Explorer and choosing “Edit”. This allows you to modify the name of the data file, force the application to start in add or modify mode and to lock various parts of the user interface.

On Android, the list of available applications on the device is constructed by finding all the pff files in the CSEntry directory and subdirectories so a pen file without a pff will not show up in the list.

Partial Save and Case Tree

It can be a pain when testing a question that is on the third form of a survey to have to reenter all of the data up to the question you are testing. We can enable partial save under the data entry options so that we can exit data entry, modify the application, and come back right to where we left off. While we are in the data entry options we can also enable the case tree on both Windows and Android to make it easier to navigate around the questionnaire while we are testing. The case tree is enabled by default on Android since Android only shows one question at a time but it is off by default on Windows. Note that on Android phones, since there is not enough space to display the case tree and the questions at the

same time so you need to tap on the big green “CS” in the top left corner of the screen to bring up the case tree.

Modifying the dictionary after collecting data

Add the variable **GHRPL household replacement status** to the dictionary and form right before number of household members. Don’t forget to modify the order in the form tree so that it is asked before number of household members. Run the application again and modify one of the existing cases. Why is the number of household members blank? Once data has been entered, you should avoid adding variables *in the middle* of a record. It is fine to add to the end of the record but adding in the middle invalidates the start positions of the existing variables. If you must add a variable in the middle of a record, you can use the reformat data tool to adjust old data files to match the new dictionary.

Group exercise

Add the labor record to the. Name this record “LABOR_REC”. How many occurrences should it have? Add the four variables from the proxy section to the new record. Create a new form for labor and add the fields onto it to create a roster. Name the roster “LABOR_ROSTER”. Use the number of household members variable as the occurrence control field to limit the size of the roster. Test the application on both Windows and on Android.

Subitems

Let’s add the **Date of Birth (DEMBTH)** to the application. In order to be able to look at both the date as an 8-digit number and look at the day, month and year individually we can create an item with subitems. Subitems are items that are made up of a subset of the digits of their parent item. Add the item for date of birth and the following subitems:

- year of birth (length 4)
- month of birth (length 2)
- day of birth (length 2)

We are putting year first then month and day because this format will work better with other CSPro features that we will see later. Click on **Layout** in the toolbar to ensure that the item and subitem overlap. We can add either the item to the form or the subitems, but not both. When we add the subitems to the form we do not need keep the same order that we have in the dictionary. On the form, we can put the day, month then the year to follow what is in the questionnaire. For our questionnaire, we will use the item rather than the subitems.

Linked Value Sets

Note that questions **GHRPL (replacement status)** and **PXYRSP (household member available)** both have the same value set (Yes/No). We could simply copy and paste the value set into both items, however, this would leave us with two copies of the same value set. If later on we change one of them

and forget to change the other, then they will be out of sync. This could cause consistency problems later on. Instead, we can create the value set for **GHRPL** and then paste it as a linked value set into **PYRSP**. With a linked value set, CSPro only stores one copy of the value set that is shared between both variables. This way if you edit the value set, the change is reflected in both items. This will become even more important later when we add translations of value sets in other languages.

Exercises

1. Add the remaining fields from the demographics section of the questionnaire to the dictionary and the household roster (**DEMEDL**, **DEMEDC**). Make sure to add the appropriate value sets. You can use a linked value set for **DEMEDC**.

Tip: When dragging items onto a form with an existing roster, drop the items *inside* the roster or you will end up with a second roster for the new item.

2. Add the fields for the employment at work module (**EMP**)
3. Add the fields for the temporary absence module (**ABS**)

Make sure to test your application on both Windows and Android.