

Session 3: CAPI Features

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

- Add question text and help text
- Add fills in question text
- Use conditions to modify question text
- Use multiple languages in question text, error messages and in the dictionary
- Use capture types to implement date pickers and multiple response questions (checkboxes)

Question Text

We can add text to each question in our survey by clicking on *CAPI Questions* in the toolbar. This allows us to enter literal question text for the questionnaire that the interviewer will read verbatim. Additionally, we can add interviewer instructions. Let's start with the first few questions in the demographics section.

For *NAME*, enter "What are the names of the members of this household, starting with the head or reference person?". For **DEMSEX** enter "Is (name) male or a female?". Run the application on Windows and then on Android to see how the question text is displayed.

In addition to the text, we can add instructions to the interviewer. For example, for **DEMAG1** we can have:

How old is (name)?

Followed by the instruction:

Write '0' if less than 1 year

To make it clear to the interviewer that this is an instruction, we use italics to distinguish it from the question. We can also use different colors and fonts as well. You can use whatever scheme you like as long as it is consistent scheme throughout the questionnaire.

If you have your question text in Word or Excel you can copy and paste into CSPro and it will preserve the formatting.

Help Text

In addition to question text, we can provide additional instructions to the interviewer as “help text”. Help text is not shown by default but can be displayed by using the F2 key on Windows or tapping the help icon next to the question text on Android.

Let’s add the following help text to the NAME field:

Include all persons living in this house who have common arrangements for cooking and dining.

Run the application on both Android and Windows and see how the help text is displayed.

Fills in Question Text

It is possible to have the question include the values of dictionary variables. For example, for **SEX**, instead of asking “Is (name) male or a female?” we can include the respondents name by using %NAME% inside the question text. At runtime, this will be replaced by the contents of the variable NAME.

Let’s change the text for DEMSEX and DEMAG1 to use %NAME% in place of “(name)”.

Let’s do the same for PXYRSP in the labor roster:

Now I would like to talk to (PPNO# NAME). Is (PPNO# NAME) available?

Since we are in the labor roster, instead of just using %NAME%, we need to use a subscript like, %NAME(LABPPNO)% to make sure that we get the correct name.

Group exercise

Add the question text, including fills, for relationship and date of birth.

Conditional Question Text

Sometimes, the question will be completely different depending on other factors and using fill variables is not sufficient to implement the differences. For example, the question text for the NAME field should really be:

What is the name of the head or reference person? – for the first row of the roster

What is the name of the next member of the household? – for the remaining rows of the roster.

We can implement this using the conditions window underneath the question text window. By default, this window contains a single empty condition. Right-clicking on the conditions lets you add a new condition. Let’s modify the first condition so that **Min Occ** and **Max Occ** are both 1 and then add a second condition where **Min Occ** is 2 and **Max Occ** is 20. Now we have two different question texts, one that will be shown for the first occurrence and the other that will be shown for the rest. Edit these two questions texts as above and test it to verify that it works.

	Min Occ	Max Occ	Condition
1	1	1	
2	2	20	

Another question that has conditional text is PXYCHK:

IS (PPNO# NAME) (HIM/HERSELF) RESPONDING TO THE QUESTIONS?

PPN# NAME is just NAME with the appropriate subscript but (HIM/HERSELF) depends on the sex of the individual. We can have two messages: one for males that uses “himself” and another for females that uses “herself”. For this we need to set the condition expression in the CAPI text instead of the occurrences:

	Min Occ	Max Occ	Condition
1	1	20	DEMSEX(LABPPNO) = 1
2	1	20	DEMSEX(LABPPNO) = 2

Logic Variables in Question Text

In the question EMPPAY we have the following question text:

In the last week, that is from [DATE] up to [DATE] did (you/name) do any work for a wage, salary or any other pay, even if only for one hour?

We could use the interview date (**GHVFDT**) as the fill for the second date but we don’t have a variable in the dictionary to use for the date from one week before the interview. Instead of creating a new dictionary variable, we can create a logic variable. Logic variables are like dictionary variables but they cannot appear on a form and are not saved to the data file. There are three types of logic variables: numeric, alpha and string. Numeric and alpha logic variables represent numeric and alphanumeric values just like numeric and alpha dictionary variables. String logic variables are used for variable length alphanumeric variables. In earlier versions of CSPro, string logic variables did not exist so you can find a lot of old CSPro applications that use alpha logic variables. With modern CSPro we recommend always using string variables rather than alpha.

To declare a variable that we can use in the question text, we add a declaration line in the PROC GLOBAL. All logic variables in CSPro must be declared before they can be used. You can get to the PROC GLOBAL by clicking on the top of the form tree when in logic view. Any variable declared in PROC GLOBAL can be used in any PROC in the application.

Let’s declare two string variables to use for the interview date and the date one week before the interview. While the dates are numeric, using strings will allow us to format the dates to use in the question text with “/” in them rather than just displaying an eight-digit number.

```
PROC GLOBAL
string interviewDate, weekBeforeInterviewDate;
```

We recommend using mixed case (aka camelCase) for logic variables to distinguish them from dictionary variables.

We can now add the fills to the question text.

In the last week, that is from %weekBeforeInterviewDate% up to %interviewDate% did %NAME% do any work for a wage, salary or any other pay, even if only for one hour?

Where should fill in the values for these two variables? We could do it anywhere *before* the field where they are displayed (**EMPPAY**) but let's fill them in as soon as we get the interview date in the postproc of **GHVFDT**.

```
PROC GHVFDT
// Create string versions of interview date and week before interview date
// to use as fills in question text
interviewDate = edit("9999/99/99",GHVFDT);
weekBeforeInterviewDate = edit("9999/99/99", dateadd(GHVFDT, -7, "d"));
```

To format the interview date with “/” in between we use the function *edit* which converts numbers to formatted strings. Edit takes two arguments, a string pattern and a number. Any “9” in the pattern is replaced by a digit from the number. Any “Z” in the pattern is replaced by a zero-filled digit from the number. By using the pattern “9999/99/99” we should get a date in “yyyy/mm/dd” format.

To compute one week before the interview date we use the function *dateadd* which adds an interval in years, months or days to a date. In this case we will add negative 7 days to the interview date to get the date one-week prior.

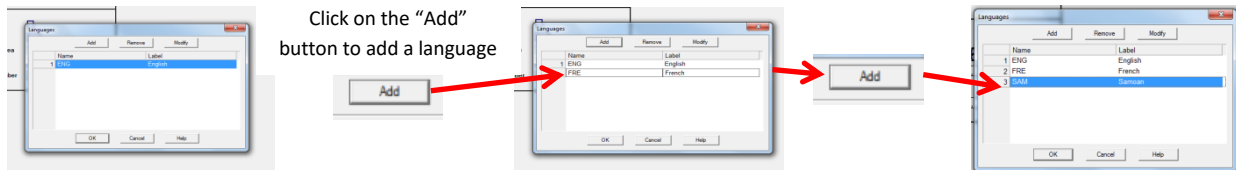
Supporting Multiple Languages

The first It is possible to have the question text, variable labels, error messages and value sets displayed in multiple languages. Language can be specified for question text, messages and for dictionary labels.

Multiple Languages for Question Text

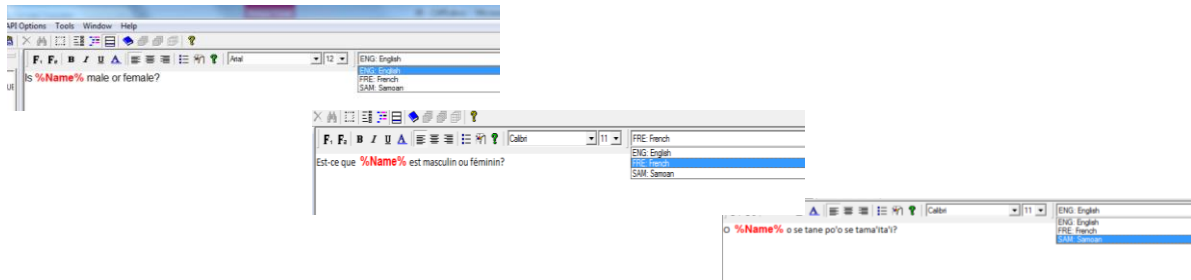
To have multiple languages for question text you first need to define the languages. This is done from the CAPI Options Menu. Select “Define CAPI Languages”

This will bring up the “Language” dialog box. Click on “Add” to add a language and then on “OK”.



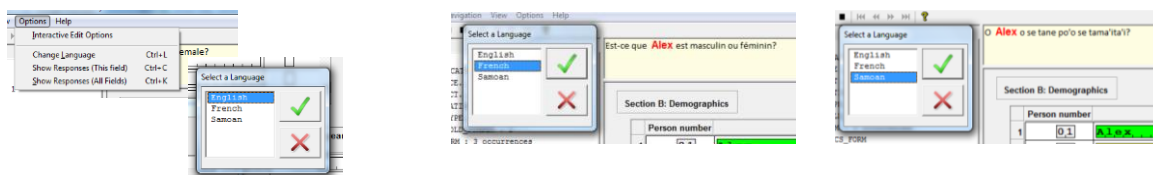
- This simply specifies the names you will be using for the languages. It has no effect on the translations.
- Language names follow the same rules as names; that is, they must be unique and cannot contain spaces. Try using abbreviations like ENG (English), SPA (Spanish), FRA (French), or POR (Portuguese).
- Language labels can contain any text to describe the language.
- During data entry, the interviewer can easily switch among languages

When creating question text, you enter the question text for each language specified. Let’s look at the question text for the “Sex” item. There is a Drop Down menu for the languages we specified in the above steps. We select the language and then enter the question text in that language.



Now that you have created the question text in the specified languages, they can be selected during data entry.

In Windows, we can go to the “Options” Menu and select “Change Language” and a menu of the languages we defined will be displayed. Click on the desired language and the CAPI text will be displayed in that language.



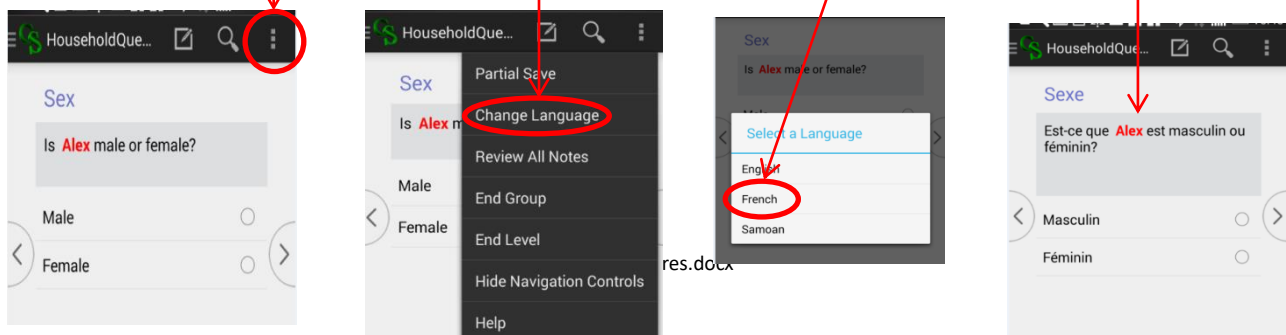
On Android:

Tap to bring up the menu

Tap “Change Language” to select the language

Tap on the language you wish to display. In this case we will select “French”.

The CAPI question is display in the selected language. In this case, it is “French”.



Multiple Languages for Dictionary Items

To have multiple language in question text you first need to define the languages. This is similar to the process done for multiple languages for CAPI questions; however, for dictionary items you define languages from the Edit menu of the dictionary.

Make sure you are in the dictionary table

Select "Edit"

Select "Languages"

- Enter the language names

Using the language drop-down at the top of the page you can cycle through the languages you defined in the above step.

This will allow you to add labels for each language specified:

N	Item Label	Item Name
	(record type)	
<input type="checkbox"/>	Province	PROVINCE
<input type="checkbox"/>	District	DISTRICT
<input type="checkbox"/>	Enumeration area	ENUMERATION_AREA
<input type="checkbox"/>	Area type	AREA_TYPE
<input type="checkbox"/>	Household number	HOUSEHOLD_NUMBER
<input type="checkbox"/>	Numéro de personne	PERSON_NUMBER
<input type="checkbox"/>	Nom	NAME
<input type="checkbox"/>	Lien de parenté	RELATIONSHIP
<input type="checkbox"/>	Sexe	SEX
<input type="checkbox"/>	l'Age	AGE
<input type="checkbox"/>	Date de naissance	DATE_OF_BIRTH
<input type="checkbox"/>	Année de naissance	YEAR_OF_BIRTH
<input type="checkbox"/>	Mois de naissance	MONTH_OF_BIRTH
<input type="checkbox"/>	Jour de naissance	DAY_OF_BIRTH
<input type="checkbox"/>	Lieu de naissance	PLACE_OF_BIRTH

N	Value Set Label	Value Set Name	Value Label
<input type="checkbox"/>	Le lien de parenté	RELATIONSHIP_VS1	
<input type="checkbox"/>			Chef de ménage
<input type="checkbox"/>			Epouse
<input type="checkbox"/>			Filles/Fille
<input type="checkbox"/>			Frer/Souer
<input type="checkbox"/>			Pere/Mere
<input type="checkbox"/>			Petit enfant
<input type="checkbox"/>			autre lien de parente
<input type="checkbox"/>			sans lien de parente

Multiple Language Messages

To add translations used in program logic, such as arguments to `errmsg()`, we can provide translations in the message file. To do this we add the translated messages to the messages tab at the bottom of the logic editor.

```
{Application 'POPSTANLFS' message file generated by CSPro}
FRA("%s is the head of household and is only %d years old. The head of household must be at least
15 years old") Le chef de menage a %d ans. Le chef de menage doit avoir au moins 15 ans.
FRA("Fix relationship") Corriger le lien de parente
FRA("Corriger l'âge") Corriger l'age
FRA("Ignore error") Ignorer l'erreur
```

On each line of the file, we put the three-digit language code followed by the English message text in parenthesis and finally the translation. Then in the logic, we use the function `tr()` to retrieve the translated text. Now when we run the application with the language set to French and encounter the error, it will show the French text.

```
PROC DEMAG1
// Ensure that the head of household is at least 15 years old.
if DEMREL = 1 and DEMAG1 < 15 then
    errmsg(tr("%s is the head of household and is only %d years old. The head
of household must be at least 15 years old"), strip(NAME), DEMAG1)
    select(tr("Fix relationship"), DEMREL,
        tr("Fix age"), DEMAG1,
        tr("Ignore error"), continue);
endif;
```

Group exercise

Add the French translations for the question text, labels, value sets and error messages for highest level of education (DEMEDL).

Multiple Response Questions (Checkboxes)

The first question in the agriculture module, **AGFCHK**, allows for multiple responses. In CSPro this kind of question is implemented using checkboxes. Checkboxes offer a friendly interface for multiple response questions by presenting a single screen with a checkbox for each option.

In CSPro, multiple response questions are implemented as alpha variables whose length is the same as the number of options that can be selected at the same time. The value set has a value for each option, which is usually a single character. The resulting value is a string containing the characters for each of the selected items.

For **AGFCHK** we have 3 options, so we need a length 3 alpha variable. We will use the following value set:

Farming or rearing animals	1
Fishing or fish farming	2
Another type of job or business	3

If the interviewer checks the boxes for all three options, the value for the variable will be the string "123". If they choose only the first and last option it will be "13". We will see later how can convert the alpha value into a series of yes/no values to simplify analysis.

Skipping Based on Checkboxes

We only want to ask question **AGFMKT** if options 1 or 2 and not option 3 are selected in the multiple response field **AGFCHK**. How can we tell in logic if an option is checked? For example, how could we tell if option 2 is selected? If the interviewer picks only option 2 it is easy since the value of **AGFCHK** will be the string "2" but if they pick both the first two options then **AGFCHK** will be "12".

In CPro logic we can use the function *pos()* which returns the position of one string within another. For example *pos("C", "CPro")* will be one, *pos("P", "CPro")* will be three and *pos("o", "CPro")* will be five. If the search string is not found, *pos()* will return zero. In our case, the "2" is not always at the same position in the string. If the interviewer chooses just option 2 then 2 will be the first character in the string but if the interviewer chooses 1 and 2 then the result will be "12" and 2 will be in position two. However, all we care about is whether or not "2" is in the result string. For this we can check the result of *pos()*. If the result of *pos()* is nonzero then "2" is in the string and if the result is zero, "2" is not in the string. So to determine if option 2 is selected we can simply check *pos("2", AGFCHK) > 0*. Similarly, to check if option 1 is selected we use *pos("1", AGFCHK) > 0* and to check if option 3 is NOT selected we use *pos("3", AGFCHK) = 0*.

Putting it all together the logic for the universe for **AGFMKT** is:

```
PROC AGFMKT
preproc
// Only ask if farming or fishing and not other were picked in AGFCHK or if
// farming/fishing was picked in AGFANY
ask if ((pos("1", AGFCHK) > 0 or pos("2", AGFCHK) > 0) and pos("3", AGFCHK) =
0) or AGFANY = 1;
```

Capture Types

If you right click on the **AGFCHK** field on the form and choose “Field Properties...” you will see that the capture type is set to check box. There are other capture types:

- Text Box: keyboard data entry
- Radio button: choose one option from many with radio button for each option
- Drop Down: choose one option from many without radio buttons.
- Combo Box: combination of keyboard input and drop down (same as drop down on Windows)
- Check Box: choose multiple responses
- Date: choose a date using a special control for picking dates

When you drag an item on the form, CSPro sets the capture type based on the value set for that item. If there is no value set when you drop the item, the capture type will be set to text box. You can always change the capture using the Field Properties dialog.

Combo Boxes

Combo boxes are useful when you have a value set that contains a valid range and a discrete code for “don’t know”. The combo box shows both the range and the code discrete code when you tap on the icon in the corner of the text box. We can use this for the day, month and year of birth. For the day of birth create a value set with a valid range from 1 to 31 and don’t know as code 99. On the form, right click on the day of birth and change the capture type to combo box and run the application to see what it looks like.

Date Controls

Let’s change the interview date to use a date control. Make sure to set the format to YYYYMMDD to match the format we are using.

Exercises

1. Add the question text to the rest of the demographic section, the proxy section, the employment at work module and the temporary absence module. Use the appropriate fills and conditions.
2. Add a new language, the language of your choice, to the question text and to the dictionary. Translate the question text, labels, value sets and error messages for the demographics section into the new language.
3. Complete the Agricultural Work and Market Orientation (AGF) module including the skips and question text. For now, leave out AGFDS and AGFOCC. We will add those questions later. Update the skips in proxy and employment at work sections to skip to the appropriate parts of the agricultural module according to the specifications.
4. Implement the working time module including all skips, question texts and fills. Use checkboxes for ACDYSM and ACDYSS. For ACHRSM and ACHRSS, create seven variables – one for each day of the week - and use ask if to only display those days that were selected in ACDYSM/ACDYSS. Use logic variables for the fills for the total hours in the question text of ACHRM and ACHRS.