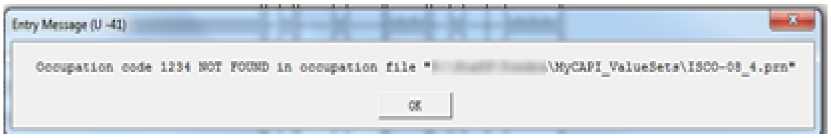
Using Lookup (External) files in CSPro

(For this example you should have a working knowledge of setting up and using a CSPro Data Dictionary and Data Entry Program.)

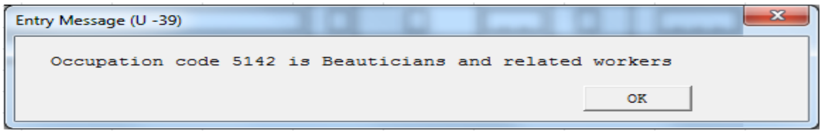
Here we describe using a lookup(external) file in in CSPro. We will use the MyCAPI_Intro application and add the item “Occupation” which will be a 4-digit occupation code. We often use the term “Look-Up” file when using external files in CSPro because the most common use for external files is to “look up” information in a “reference” file, that is, a file containing reference data that is not part of our main data entry file. This example uses CAPI, however, lookup(external) files work the same way for all modules of CSPro.

In this example, we will “look up” an occupation code in a file of occupation codes. If the occupation code is not there, we display an error message the occupation was not found:



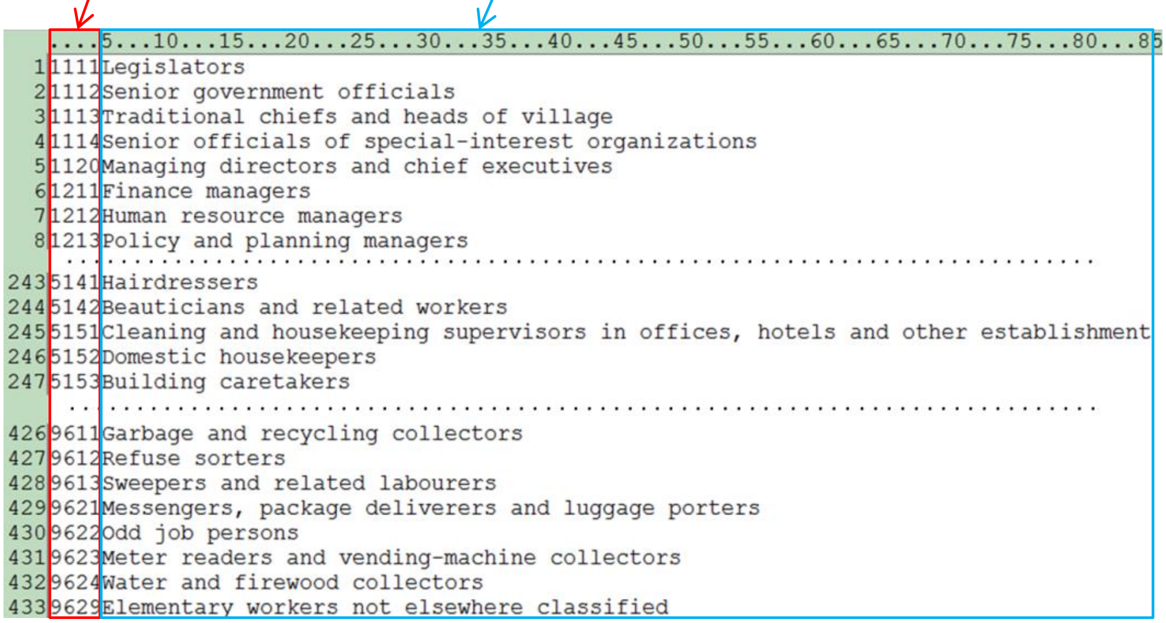
Occupation code 1234 NOT FOUND in occupation file "[redacted]\MyCAPI_ValueSets\ISCO-08_4.prn"

If the occupation code is found, we display the description of the occupation:



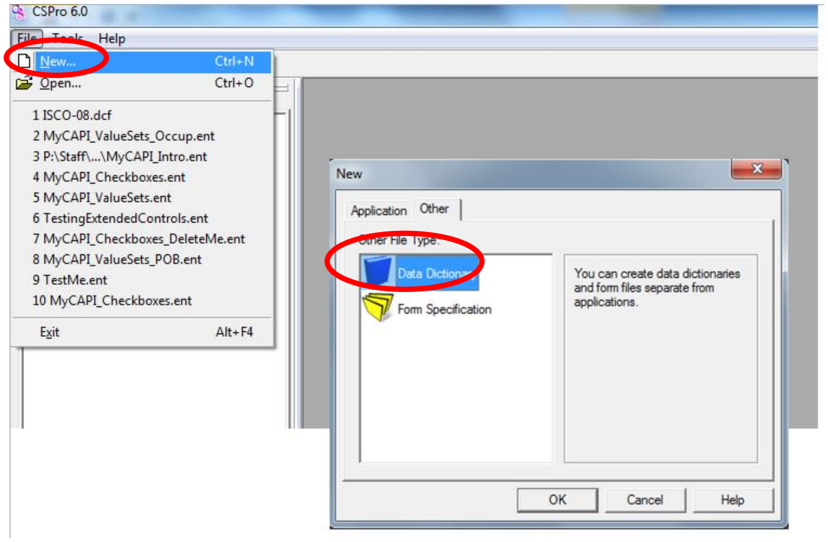
Occupation code 5142 is Beauticians and related workers

We begin with the occupation code file. The file we will be using in this example is ISCO-08_4.prn It consists of a four digit occupation code and an 80 digit description of the occupation.



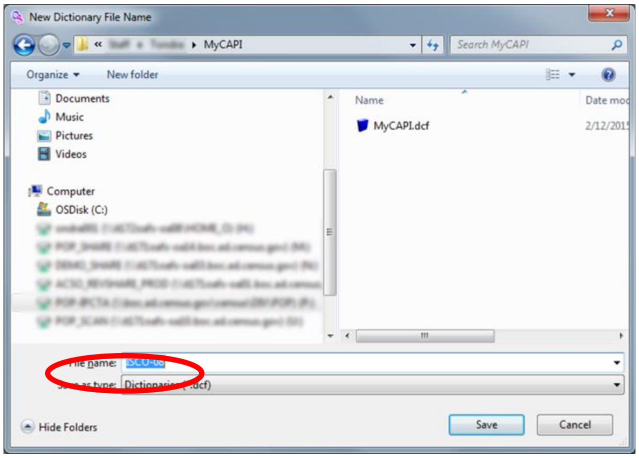
```
   ....5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85
  11111Legislators
  21112Senior government officials
  31113Traditional chiefs and heads of village
  41114Senior officials of special-interest organizations
  51120Managing directors and chief executives
  61211Finance managers
  71212Human resource managers
  81213Policy and planning managers
   ...................................................................................
2435141Hairdressers
2445142Beauticians and related workers
2455151Cleaning and housekeeping supervisors in offices, hotels and other establishment
2465152Domestic housekeepers
2475153Building caretakers
   ...................................................................................
4269611Garbage and recycling collectors
4279612Refuse sorters
4289613Sweepers and related labourers
4299621Messengers, package deliverers and luggage porters
4309622Odd job persons
4319623Meter readers and vending-machine collectors
4329624Water and firewood collectors
4339629Elementary workers not elsewhere classified
```

To use this file as an external file we will need to create a CSPro data dictionary that describes the data in the file. Select File/New from the Main menu, select the “Other” tab, click on “Data Dictionary”.



You can create data dictionaries and form files separate from applications.

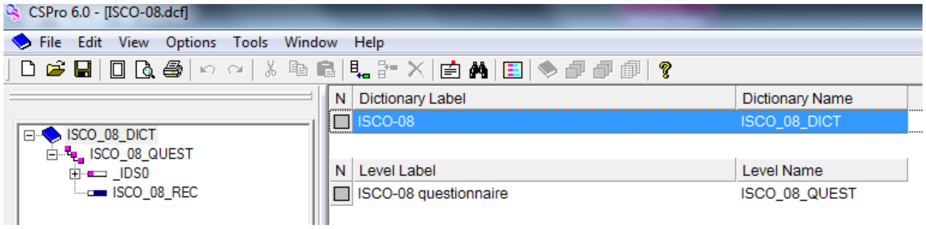
CSPro will ask for the new dictionary file name. Navigate to the folder containing your application and enter a name for you dictionary. I will give the dictionary the name ISCO-08.



File name: ISCO-08

Save as type: Dictionaries (*.dcf)

CSPro creates a dictionary template:

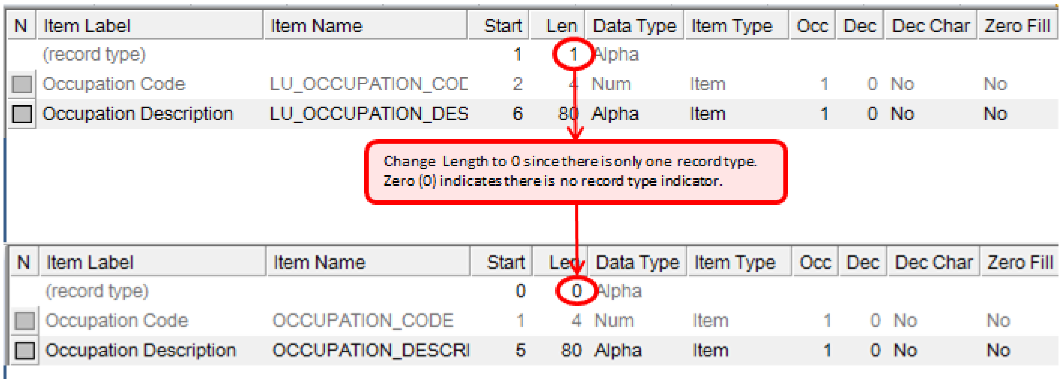


N	Dictionary Label	Dictionary Name
□	ISCO-08	ISCO_08_DICT

N	Level Label	Level Name
□	ISCO-08 questionnaire	ISCO_08_QUEST

We will modify this template so the dictionary describes the occupation code file by

1. Changing **Item Label** ISCO-08 to “Occupation Code”.
2. Changing **Item Nam**e ISCO_08_ID to LU_OCCUPATION_CODE.
3. Adding the item “Occupation Description” to the ISCO_08_REC record.
    a. The **Item La**bel is “Occupation Description”
    b. The **Item Name** is “LU_OCCUPATION_DESCRIPTION.
    *(I am using the prefix “LU_” to indicate that this item is from the **L**ook**U**p file)*
4. Change the length of “(record type)” to zero(0). Since we have only one record type we do not need a record type indicator. Entering a zero(0) here tells CSPro that we do not need a record type indicator.



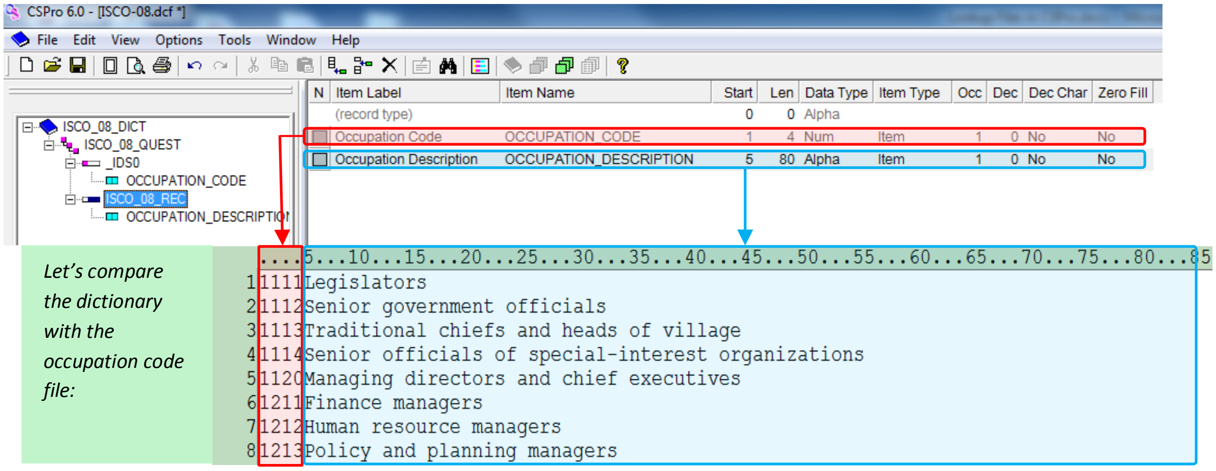
N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		1	1	Alpha					
▢	Occupation Code	LU_OCCUPATION_COD	2	4	Num	Item	1	0	No	No
▢	Occupation Description	LU_OCCUPATION_DES	6	80	Alpha	Item	1	0	No	No

Change Length to 0 since there is only one record type.
Zero (0) indicates there is no record type indicator.

N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		0	0	Alpha					
▢	Occupation Code	OCCUPATION_CODE	1	4	Num	Item	1	0	No	No
▢	Occupation Description	OCCUPATION_DESCRI	5	80	Alpha	Item	1	0	No	No

5. Change the length of **OCCUPATION_CODE** to **4** so it matches the occupation code in the file.
6. Change the length of **OCCUPATION_DESCRIPTION** to **80** so it matches the description in the file
7. Change the data type of **OCCUPATION_DESCRIPTION** to **Alpha**.

The occupation code data dictionary (ISCO-08.dcf) will look as follows:



N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		0	0	Alpha					
▢	Occupation Code	OCCUPATION_CODE	1	4	Num	Item	1	0	No	No
▢	Occupation Description	OCCUPATION_DESCRIPTION	5	80	Alpha	Item	1	0	No	No

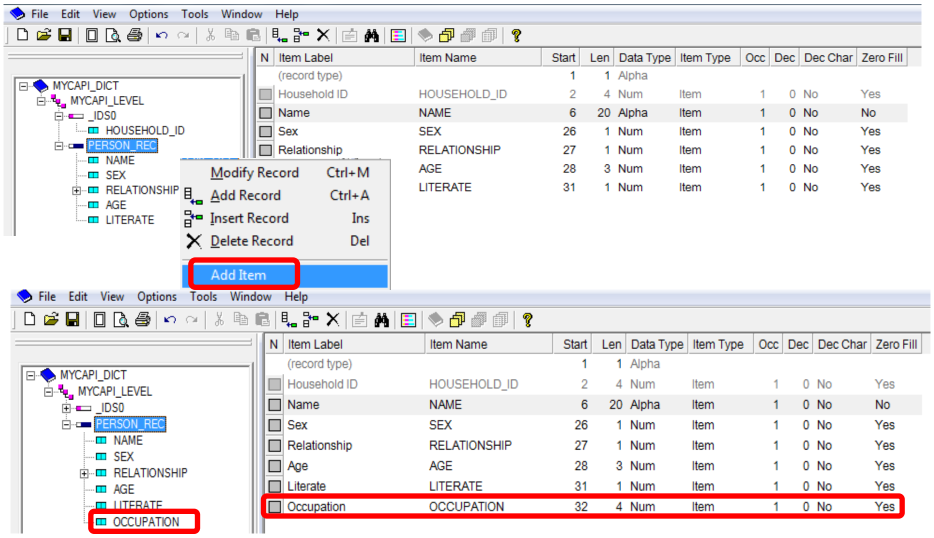
Let’s compare the dictionary with the occupation code file:

```
 ....5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85
11111Legislators
21112Senior government officials
31113Traditional chiefs and heads of village
41114Senior officials of special-interest organizations
51120Managing directors and chief executives
61211Finance managers
71212Human resource managers
81213Policy and planning managers
```

Now that that we have the dictionary for our "look up" file we will add OCCUPATION to our MyCAPI data dictionary. In our logic, CSPro will check to see if the occupation code is valid, that is, is it in the lookup file. If it is, our application will issue a message with the occupation description. If the occupation code is not found in the occupation code file, our application will issue a message stating that "Occupation #### was not found in the occupation file".

We add "Occupation" to our MyCAPI application data dictionary:

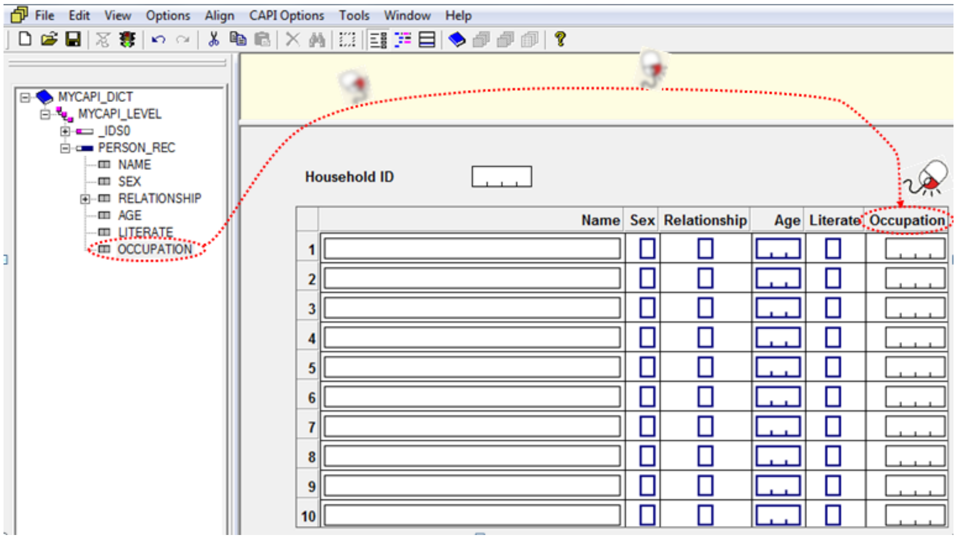
1. Open the MyCAPI application
2. Go to the dictionary panel by either clicking on the dictionary button on the tool bar or the dictionary tab .
3. Add OCCUPATION item to the PERSON_REC



N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		1	1	Alpha					
	Household ID	HOUSEHOLD_ID	2	4	Num	Item	1	0	No	Yes
	Name	NAME	6	20	Alpha	Item	1	0	No	No
	Sex	SEX	26	1	Num	Item	1	0	No	Yes
	Relationship	RELATIONSHIP	27	1	Num	Item	1	0	No	Yes
		AGE	28	3	Num	Item	1	0	No	Yes
		LITERATE	31	1	Num	Item	1	0	No	Yes

N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		1	1	Alpha					
	Household ID	HOUSEHOLD_ID	2	4	Num	Item	1	0	No	Yes
	Name	NAME	6	20	Alpha	Item	1	0	No	No
	Sex	SEX	26	1	Num	Item	1	0	No	Yes
	Relationship	RELATIONSHIP	27	1	Num	Item	1	0	No	Yes
	Age	AGE	28	3	Num	Item	1	0	No	Yes
	Literate	LITERATE	31	1	Num	Item	1	0	No	Yes
	Occupation	OCCUPATION	32	4	Num	Item	1	0	No	Yes

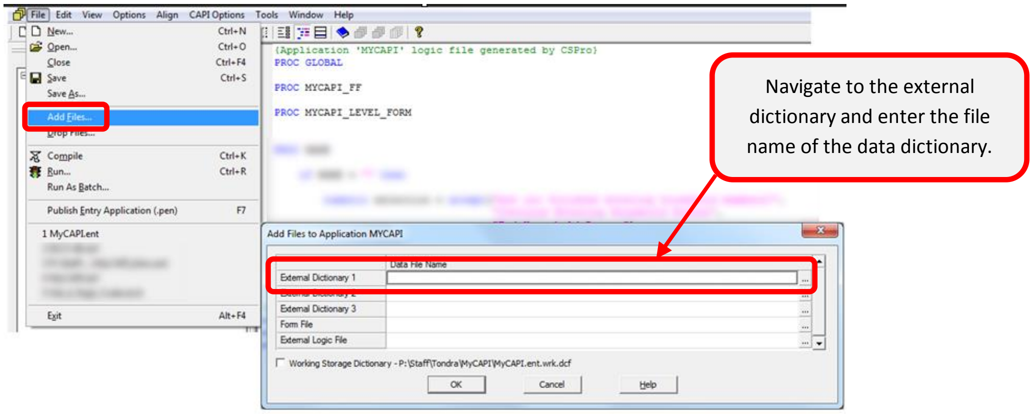
4. Add occupation to the form by dragging the OCCUPATION item from the dictionary to the form



Household ID

	Name	Sex	Relationship	Age	Literate	Occupation
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

5. We now add the occupation code dictionary to our application
    a. Go to the dictionary tab by clicking on the dictionary button on the tool bar or on the dictionary tab Files Dicts Forms .
    b. We add the external dictionary by clicking on on "File", "Add Files". This brings up the dialog box to enter the name of the file. **It is important to use "Add Files" rather than open** because "**Add Files**" adds the dictionary file as an **External File** whereas "**Open**" does not.



```
{Application 'MYCAPI' logic file generated by CSPro}
PROC GLOBAL

PROC MYCAPI_FF

PROC MYCAPI_LEVEL_FORM
```

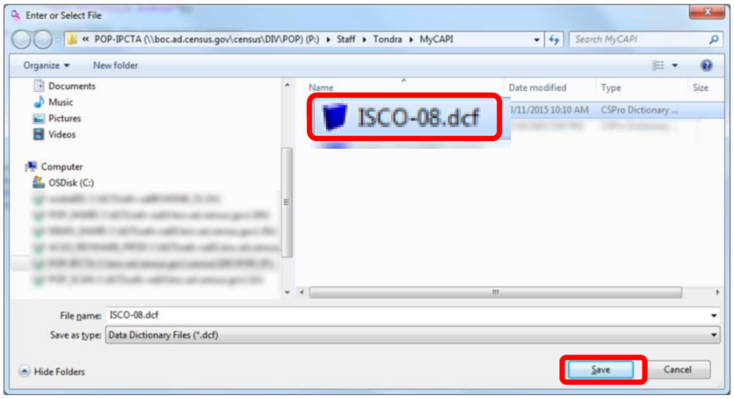
Navigate to the external dictionary and enter the file name of the data dictionary.

Add Files to Application MYCAPI

	Data File Name
External Dictionary 1	
External Dictionary 2	
External Dictionary 3	
Form File	
External Logic File	

☐ Working Storage Dictionary - P:\Staff\Tondra\MyCAPI\MyCAPI.ent.wrk.dcf

    c. We navigate to our application folder,
        i. select the external dictionary and click "Save".


Enter or Select File

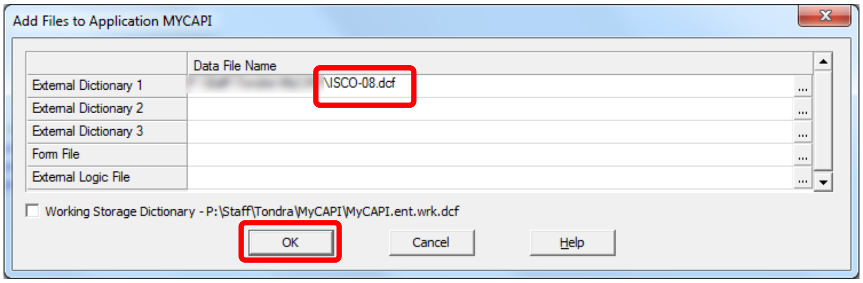
« POP-IPCTA (\\boc.ad.census.gov\census\DIV\POP) (P:) ▸ Staff ▸ Tondra ▸ MyCAPI

Name	Date modified	Type	Size
ISCO-08.dcf	1/11/2015 10:10 AM	CSPro Dictionary ...	

File name: ISCO-08.dcf

Save as type: Data Dictionary Files (*.dcf)

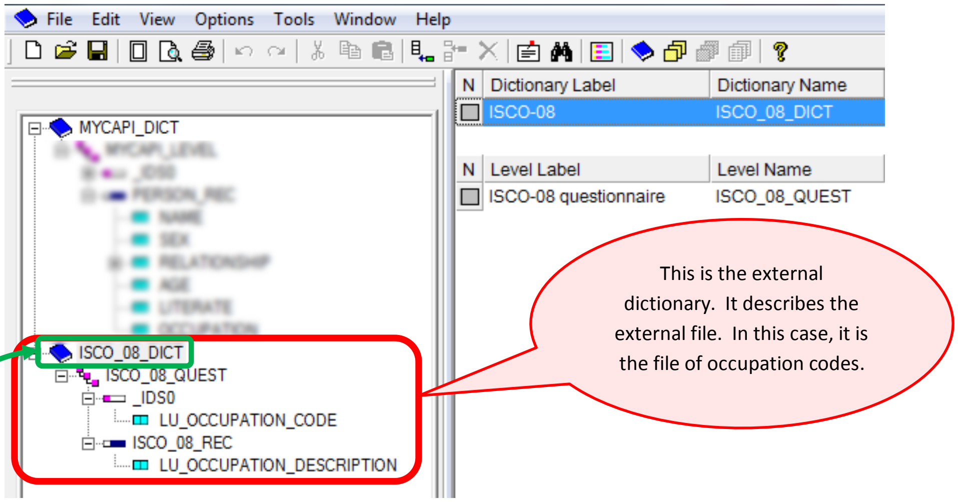
        ii. Verify this is the correct external dictionary. In this case, we are adding the occupation code data dictionary ISCO-08.dcf. If correct, click "OK".


Add Files to Application MYCAPI

	Data File Name
External Dictionary 1	\ISCO-08.dcf
External Dictionary 2	
External Dictionary 3	
Form File	
External Logic File	

☐ Working Storage Dictionary - P:\Staff\Tondra\MyCAPI\MyCAPI.ent.wrk.dcf

CSPro will add the dictionary to the application as an external file.



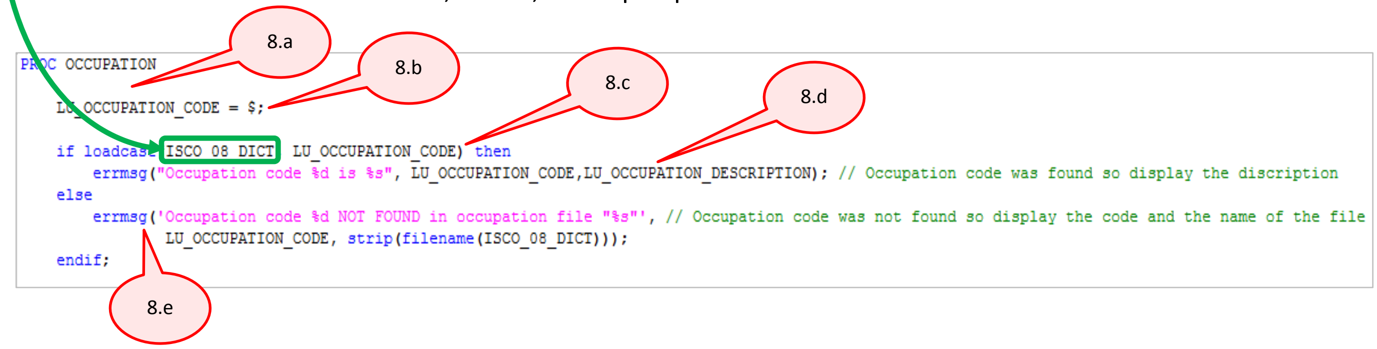
N	Dictionary Label	Dictionary Name
□	ISCO-08	ISCO_08_DICT

N	Level Label	Level Name
□	ISCO-08 questionnaire	ISCO_08_QUEST

This is the external dictionary. It describes the external file. In this case, it is the file of occupation codes.

6. The occupation data dictionary, ISCO-08.dcf, is now part of the application as an "**External File**".

7. We are now ready to program the check of the occupation code it. This is done after OCCUPATION has been entered, that is, in the postproc of the OCCUPATION item…



```
PROC OCCUPATION

    LU_OCCUPATION_CODE = $;

    if loadcase(ISCO_08_DICT, LU_OCCUPATION_CODE) then
        errmsg("Occupation code %d is %s", LU_OCCUPATION_CODE,LU_OCCUPATION_DESCRIPTION); // Occupation code was found so display the discription
    else
        errmsg('Occupation code %d NOT FOUND in occupation file "%s"', // Occupation code was not found so display the code and the name of the file
                LU_OCCUPATION_CODE, strip(filename(ISCO_08_DICT)));
    endif;
```

8. Let's examine this code:

    a. This is in the postproc of OCCUPATION. *(If neither preproc nor postproc is specified the default is postproc.)*

    b. The **OCCUPATION** item contains the occupation code that has been entered. **OCCUPATION** is represented by "$". *(Remember, in CSPro, $ can substitute for the name of the item of the current proc.)* The value of occupation needs to be given to the lookup file so the "loadcase" function can search for that value in the lookup file. After this assignment, LU_OCCUPATION_CODE contains the value entered for OCCUPATION.

    c. The loadcase function searches the external file for the value contained in LU_OCCUPATION_CODE (which at this point contains the value that was given to it by the assignment **LU_OCCUPATION_CODE = $;** making it same value that was entered for OCCUPATION) . The parameters for the loadcase(parameter-i, parameter-ii) are:

        i. The name of the dictionary describing the external file (In this example, it is the occupation code dictionary (ISCO_08_DICT)).

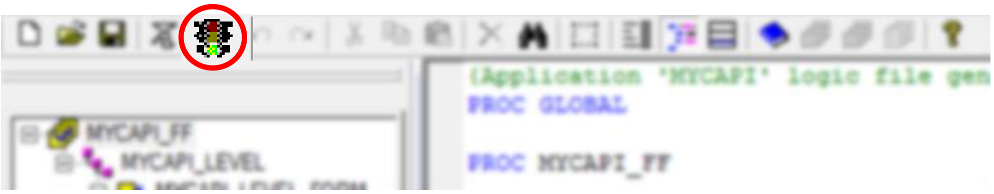
        ii. An item list containing the ID item(s) of the external file. CSPro will search this(these) item(s) to find a match (in this example the ID item is LU_OCCUPATION)

    d. If the occupation code is found in the external file, the **loadcase** function returns "True". In our example, we then issue an error message showing the occupation code and the description.

    e. If the occupation is not found in the external file, the **loadcase** function returns "False". In our example, we then issue an error message showing the occupation code and the name of the file containing our occupation codes.

9. Let's run the program. We will only look at the parts related to OCCUPATION.

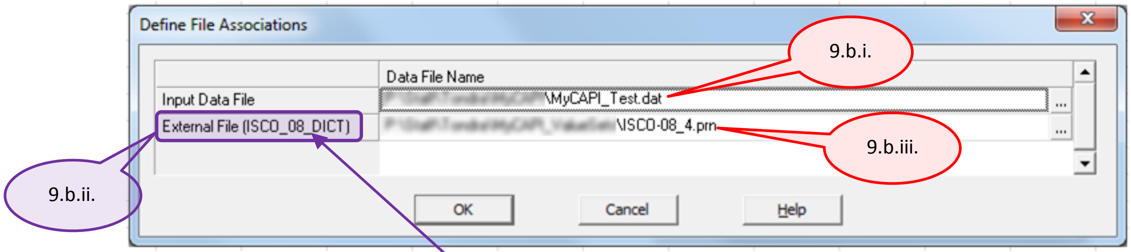
    a. Run the program by clicking on the traffic light icon on the tool bar



```
{Application 'MYCAPI' logic file gen
PROC GLOBAL

PROC MYCAPI_FF
```

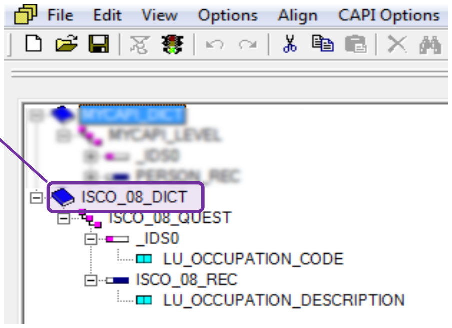
b. The “Define File Associations” dialogue box will appear.



Define File Associations

	Data File Name
Input Data File	[redacted]\MyCAPI_Test.dat
External File (ISCO_08_DICT)	[redacted]\ISCO-08_4.prn

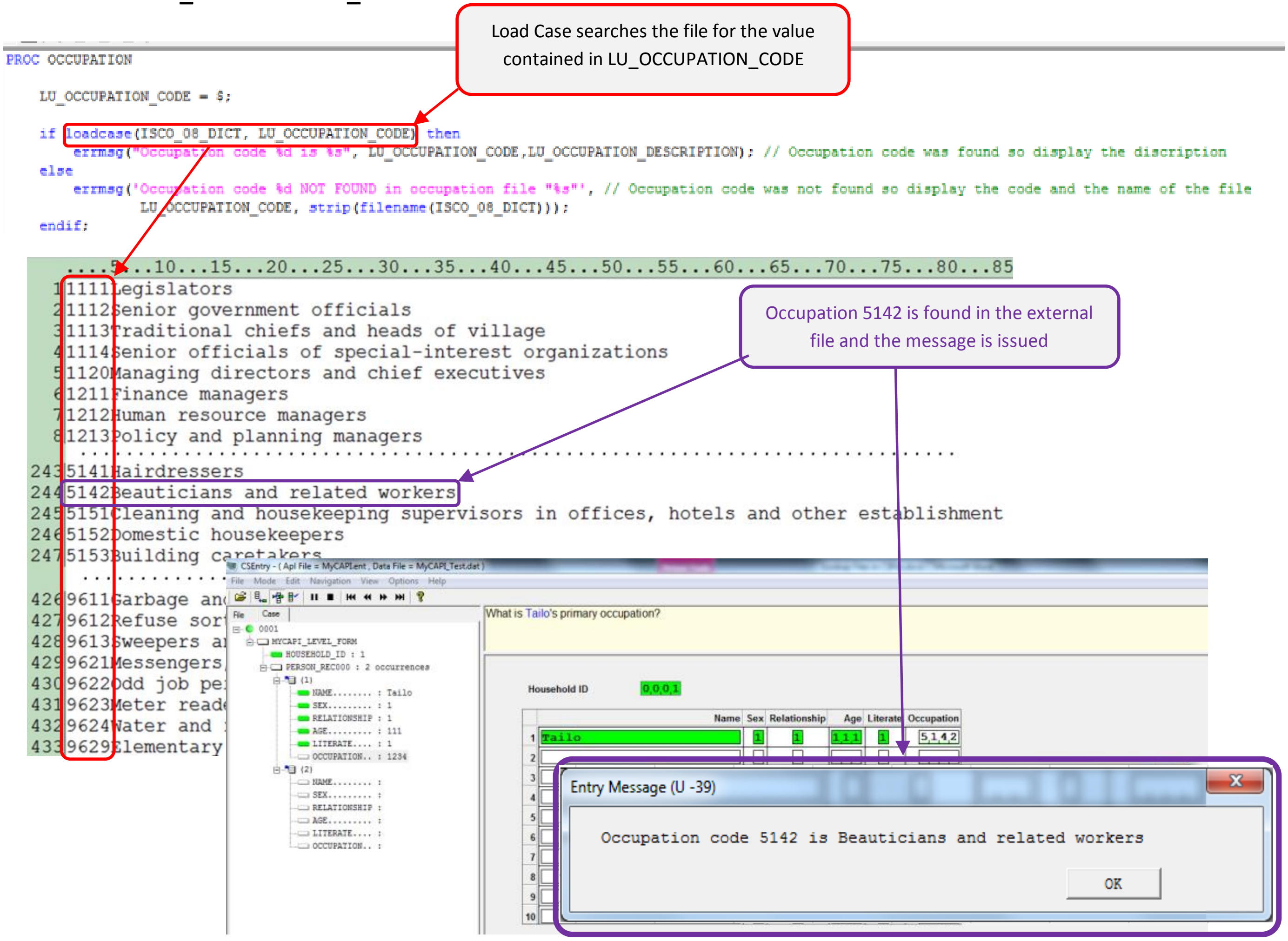
i. “Input Data File” is the name of the file that contains the data we are entering. Here I am calling it MyCAPI_Test.dat



ii. This is the name of the **data dictionary** that describes our data file. CSPro gets the name from the external dictionary(ies) defined in the application’s data dictionary section.

iii. This is the name of the file containing the external data. ***It is not the name of the data dictionary, it is the name of file the data dictionary describes.*** It is a separate file from the data file. In our example we are using ISCO-08_4.prn which is our reference file containing the 4-digit ISCO (occupation) codes.

c. Once we have entered the file names in the Define File Associations" dialogue box we click "OK" and begin entering data. For this example, I will go directly to the **OCCUPATION** item and enter **5142** for the occupation. Our program places the value entered for OCCUPATION in the ID item (**LU_OCCUPATION_CODE**) of our external file. The loadcase function then searches the file for the value contained in **LU_OCCUPATION_CODE**.



Load Case searches the file for the value contained in LU_OCCUPATION_CODE

```
PROC OCCUPATION

   LU_OCCUPATION_CODE = $;

   if loadcase(ISCO_08_DICT, LU_OCCUPATION_CODE) then
      errmsg("Occupation code %d is %s", LU_OCCUPATION_CODE,LU_OCCUPATION_DESCRIPTION); // Occupation code was found so display the discription
   else
      errmsg('Occupation code %d NOT FOUND in occupation file "%s"', // Occupation code was not found so display the code and the name of the file
            LU_OCCUPATION_CODE, strip(filename(ISCO_08_DICT)));
   endif;
```

```
    ....5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85
  11111Legislators
  21112Senior government officials
  31113Traditional chiefs and heads of village
  41114Senior officials of special-interest organizations
  51120Managing directors and chief executives
  61211Finance managers
  71212Human resource managers
  81213Policy and planning managers
    ...........................................................................
2435141Hairdressers
2445142Beauticians and related workers
2455151Cleaning and housekeeping supervisors in offices, hotels and other establishment
2465152Domestic housekeepers
2475153Building caretakers
    ...........
4269611Garbage and
4279612Refuse sort
4289613Sweepers ar
4299621Messengers,
4309622Odd job pe
4319623Meter reade
4329624Water and
4339629Elementary
```

Occupation 5142 is found in the external file and the message is issued

CSEntry - (Apl File = MyCAPI.ent , Data File = MyCAPI_Test.dat)

```
0001
  MYCAPI_LEVEL_FORM
    HOUSEHOLD_ID : 1
    PERSON_REC000 : 2 occurrences
      (1)
        NAME........ : Tailo
        SEX......... : 1
        RELATIONSHIP : 1
        AGE......... : 111
        LITERATE.... : 1
        OCCUPATION.. : 1234
      (2)
        NAME........ :
        SEX......... :
        RELATIONSHIP :
        AGE......... :
        LITERATE.... :
        OCCUPATION.. :
```

What is Tailo's primary occupation?

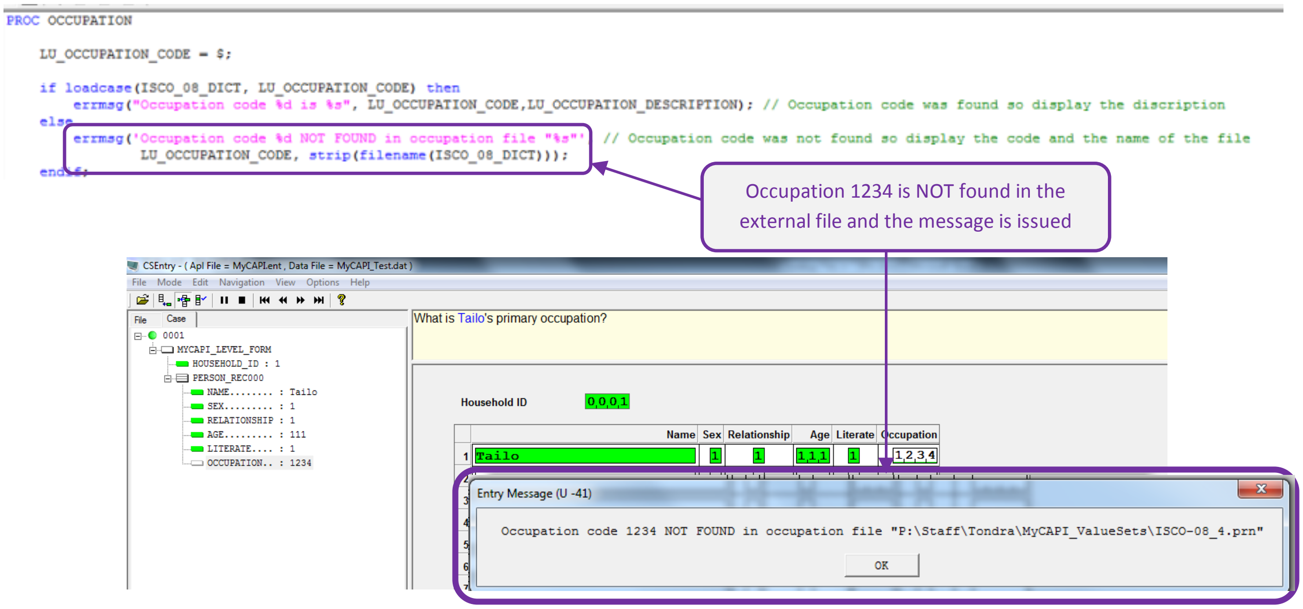
Household ID 0,0,0,1

	Name	Sex	Relationship	Age	Literate	Occupation
1	Tailo	1	1	1,1,1	1	5,1,4,2

Entry Message (U -39)

Occupation code 5142 is Beauticians and related workers

d. I will enter **1234** for the occupation. Our program places the value entered for OCCUPATION in the ID item of our external file. The **loadcase** function then searches the external file for a match to this value. This time it does not find an occupation code **1234**. The program issues the message that the occupation code was not found.



```
PROC OCCUPATION

    LU_OCCUPATION_CODE = $;

    if loadcase(ISCO_08_DICT, LU_OCCUPATION_CODE) then
        errmsg("Occupation code %d is %s", LU_OCCUPATION_CODE,LU_OCCUPATION_DESCRIPTION); // Occupation code was found so display the discription
    else
        errmsg('Occupation code %d NOT FOUND in occupation file "%s"', // Occupation code was not found so display the code and the name of the file
                LU_OCCUPATION_CODE, strip(filename(ISCO_08_DICT)));
    endif;
```

Occupation 1234 is NOT found in the external file and the message is issued

CSEntry - (Apl File = MyCAPI.ent , Data File = MyCAPI_Test.dat)

```
0001
  MYCAPI_LEVEL_FORM
    HOUSEHOLD_ID : 1
    PERSON_REC000
      NAME........ : Tailo
      SEX......... : 1
      RELATIONSHIP : 1
      AGE......... : 111
      LITERATE.... : 1
      OCCUPATION.. : 1234
```

What is Tailo's primary occupation?

Household ID 0,0,0,1

	Name	Sex	Relationship	Age	Literate	Occupation
1	Tailo	1	1	1,1,1	1	1,2,3,4

Entry Message (U -41)

```
Occupation code 1234 NOT FOUND in occupation file "P:\Staff\Tondra\MyCAPI_ValueSets\ISCO-08_4.prn"
```