

Exercise 3: Create a value-label pair from external data

At the end of this exercise you should be able to:

- a. Create an EpiData two-variable dataset from a text file
- b. Use an external two-variable dataset to create a label block internally or externally

In the previous Exercise we worked with small label blocks of not more than ten category levels. These are quickly created directly in the data form. Often we encounter the need for larger standard label blocks. This might be a list of administrative units in a country and it may have hundreds of levels. Such lists might be publicly available on the internet from the government and they can come in different formats such as text files or spreadsheets. In order to standardize as much as possible for our data collection, we should take recourse to official lists whenever possible, rather than inventing our own parallel system. If we can get such a list into a formatted file for EpiData, it might also be used in more than a single project.

In this Exercise, we provide a spreadsheet which contains two variables, the name of a given tuberculosis laboratory paired with its code, not dissimilar to a list of communities paired with their zip codes.

Formatting the external file

The first thing to know is the file type, and into which format it should be brought. We have a spreadsheet `a_ex03_namecode.xls` with two columns:

	A	B
1	Awuna	ML_J
2	Beitbridge	MS_D
3	Bindura	MC_A
4	Binga	MN_G
5	Birchenough	ML_M
6	Bonda	ML_I
7	Brunapeg	MS_G
8	Chasit...	MM...

You remember how the label block for the field `sex` was set up:

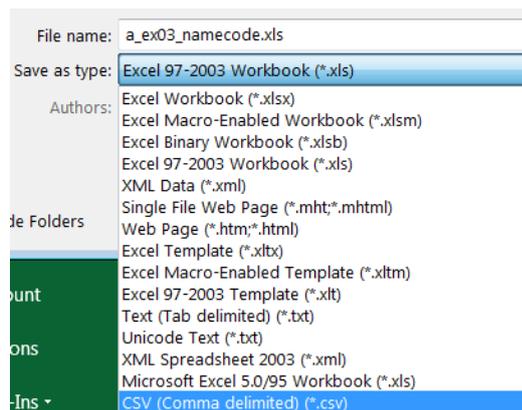
Value	Label
1	Female
2	Male
9	Not recorded

On the left is the Value that is written into the field and on the right the Label that provides the meta-data. The setup is thus the Value presenting a code on the left and the Label on the right. With short lists there is no issue because our eye captures quickly both Value and Label, but how about a very long list? How do we search a list of communities and their zip codes? People tend to know the name of a community, but not their zip codes. We are also used to search alphabetically by name not by zip code, yet we wish the code to be the value to be written into the field, not the name. EpiData makes it easy for us: we can search substrings, and it will find them in either column! If we start typing `birch...` in EpiData EntryClient, the software will

go to the first appearance of these letters be they in the left or the right column. Therefore, what we might do preferably, is to put the code of the laboratory into the left column and the name of the laboratory into the right and sort alphabetically by the name of the laboratory (not its code). This way the laboratory code is the value that is written into the field and the laboratory name is its label. Therefore invert to get:

	A	B
1	ML_J	Awuna
2	MS_D	Beitbridge
3	MC_A	Bindura
4	MN_G	Binga
5	ML_M	Birchenough
6	ML_I	Bonda
7	MS_G	Brunapeg
8	MW_J	Chegutu
9	MV_I	Chikombedzi
10	MC_H	Chimhanda

EpiData cannot read a spreadsheet file with an *.xls extension, but it reads text files with a *.txt extension or the common text standard of comma-delimited *.csv files. Conversely, most software, including proprietary Excel® or the non-proprietary LibreOffice Calc spreadsheets can save a file as a *.csv file. Use thus Save as (in Excel®) in search for the appropriate file type:



A word on delimiters: we do need a delimiter. A delimiter is an element that separates two variables. In normal text, a space is the delimiter separating two words. For our purpose, neither spaces nor tabs are good separators because they could be part of a variable. Indeed, the name for one laboratory is Collin Saunders. If we had a space as separator, these two components would erroneously be interpreted as two variables. A comma as in *.csv is better, but it is not necessarily fail-safe, best would be semi-colon delimited: it is easy to see and it is rarely used. In any case, in our 95 records (please visualize them in your text editor):

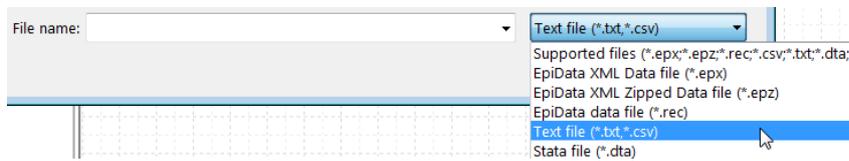
```
ML_J, Awuna
MS_D, Beitbridge
MC_A, Bindura
MN_G, Binga
ML_M, Birchenough
ML_I, Bonda
MS_G, Brunapeg
MW_J, Chegutu
MV_I, Chikombedzi
MC_H, Chimhanda
```

there are no commas apart from the desired ones to separate the two variables. We almost have now the format we require to make an EpiData *.epx file out of the *.csv file. **Important Note:** EpiData Manager interprets the first line as header but because the file does not have headers, the first line will erroneously be cut off. Therefore, we add a header to the *.csv file as follows:

```
labcode,labname
ML_J,Awuna
MS_D,Beitbridge
MC_A,Bindura
MN_G,Binga
```

Creating an EpiData *.epx file from a *.csv file

Open EpiData Manager and File | Import file ..., and select the correct file type:



Pick the a_ex03_namecode.csv file. This will open the dialog window:



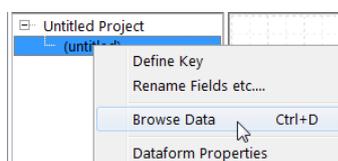
Note here the helpful options should you expect duplicates in Field Name – Field Value Labels. We can leave the defaults as they are and click OK. This gives us an Untitled Project with an (untitled) data form:



At the bottom of the screen we see that we have correctly 95 records:



Whether or not prompted to save, we save the file as a_ex03_namecode.epx. Right-clicking the data form (or with CTRL+D):

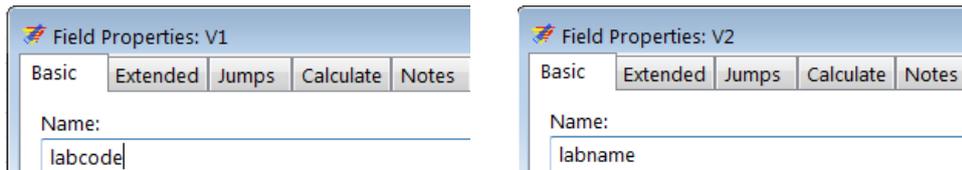


we browse to view the data set:

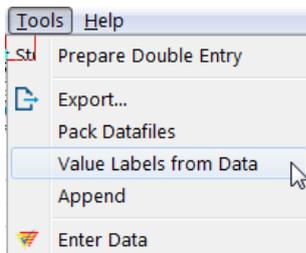


Record No:	V1	V2
1	ML_J	Awuna
2	MS_D	Beitbridge
3	MC_A	Bindura
4	MN_G	Binga

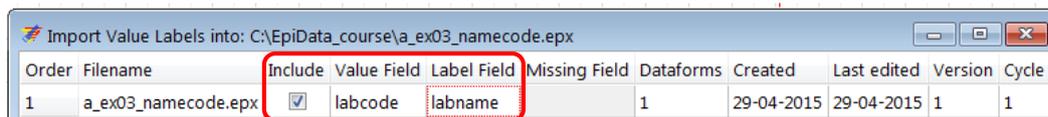
Closing and clicking on the first, and then on the second field respectively we can rename the two field names to labcode and labname to have a bit more meaningful information:



To make a label block set of the file go to Tools | Value Labels from Data:

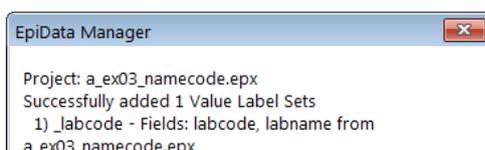


and pick the file itself, ensuring that Include is ticked and both Value Field and Label Field are correctly selected:



Order	Filename	Include	Value Field	Label Field	Missing Field	Dataforms	Created	Last edited	Version	Cycle
1	a_ex03_namecode.epx	<input checked="" type="checkbox"/>	labcode	labname		1	29-04-2015	29-04-2015	1	1

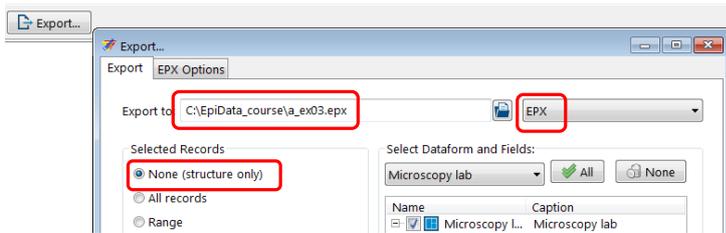
We get confirmation that the Value Label Set was created:



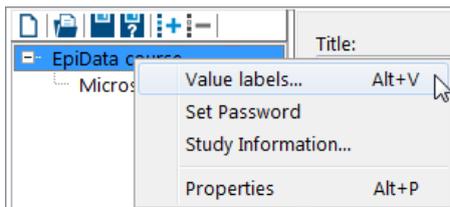
Save and close.

Updating the EpiData data form

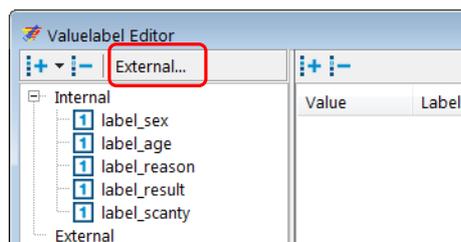
The first thing to do is to open the file a_ex02 .epx from Exercise 2 and Export the structure only to an EPX file we name a_ex03 .epx:



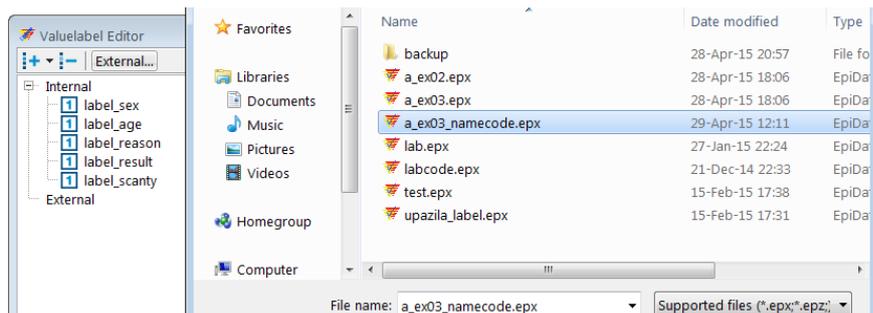
We then open the thus newly created `a_ex03.epx`. If we right-click on the project name EpiData Course we get a menu line Value labels (**ALT+V**):



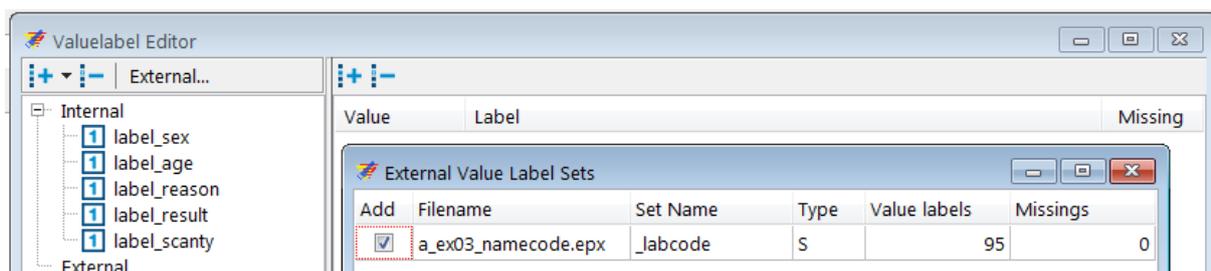
This gives us the Valuelabel Editor:



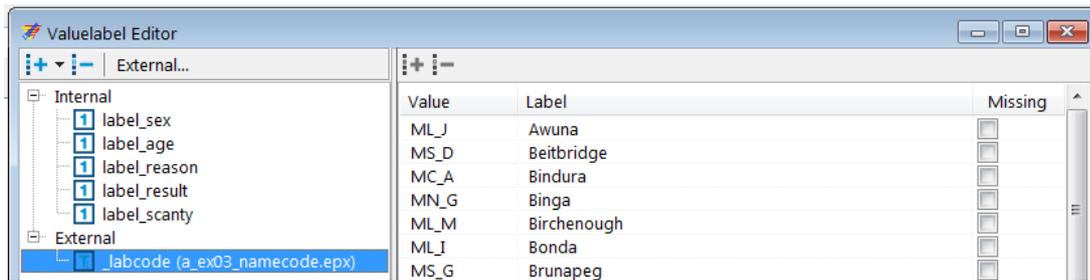
which allows the use of an external file as a label block. Using this approach (you will also look for an alternative to incorporate it), we can keep the label block externally. As long as it is in the same folder, the Manager / EntryClient will access it. Click and select the file:



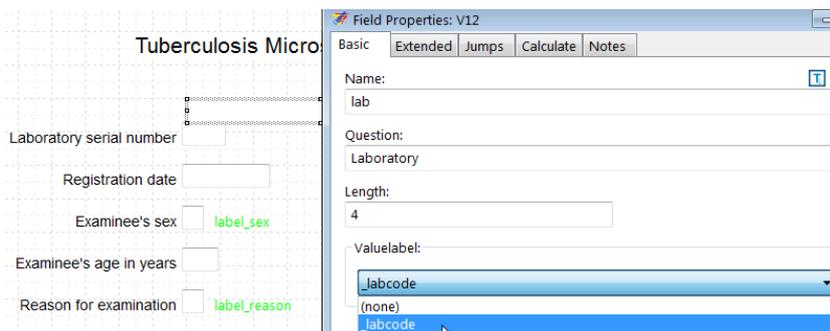
We are shown the name of the file, what the Set Name for the label block is (`_labcode`), that it is a string field with 95 Value labels:



We have now an external label block available and its content is visualized by clicking on it:



On the data entry form, we add a new string variable lab of length 4 with the field label Laboratory and add the now available Valuelabel:



Define the Extended as usual, save and test in EpiData EntryClient. Note particularly in EntryClient that for searching you start typing the name of a Field label and the cursor jumps to it.

Tasks:

- o Remove the external label block.*
- o Use the a_ex03_namecode.epx file and use it internally as a label block.*
- o Discuss advantages and disadvantages of External versus Internal label blocks.*