Exercise 5: Using an external file for Label blocks

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

a. Prepare a spreadsheet file to be saved as a CSV text formatted file acceptable for import into other database software, including an EpiData REC file
b. Know how to prepare a QES and CHK file that accept an imported text file and can be used as an external label block
c. Know how to access an external REC file from a CHK file for use as a label block

We have used label blocks to display Field value – Value label pairs. In our example, there were never more than ten values. We could have done with a few more, perhaps up to 20, perhaps even a few more, but there will certainly be an upper limit for the `CHK` file, we couldn't (or surely wouldn't be trying to) write thousands of lines into it – the `CHK` file has an upper limit: neither the `QES` file nor the `CHK` file can exceed the size of 64 KB (close to 66,000 key strokes is a lot of leverage). But apart from the file size limitations, it is also **inefficient** and **error-prone to write very large label blocks** into the Check file.

A quite common situation is that we have a list of districts or other administrative units to pick from and such lists can be quite long. As a general rule, one should always, whenever possible, use official lists and not making them up by one self. The latter is last resort if there is really no official list available. Such lists come in various formats and we will have to adapt them to our requirements. The common standard across a multitude of database formats are text files (not word processing files, they are not text files). Text files are the bare-bone elements of anything done on a computer: any key stroke on a computer can be expressed as a sequence of 8 bits (1 byte). In a text file, each keystroke (such as a letter) will take 1 byte, at least in the conventional standard (things are changing). Text files (traditionally in ASCII [*American Standard Code for Information Interchange*], now often extended to UTF-8 [*Unicode Transformation Format-8*]) will be accepted for import by virtually any decent software and can also be produced (exported to) by virtually any decent software. What is furthermore needed is a *delimiter*. A delimiter separates different things, two words for instance should be separated by a delimiter. In a standard text as this one here, the delimiter between two words is a space. In a data set, the space is not a good choice for a delimiter because there could be text fields that themselves have several words in one field, and this would then promptly lead to ambiguity and indeed errors. Tab stops are also not good delimiters because text editors often convert them into a series of spaces. That leaves the comma and the semicolon. Personally, we give preference to the semicolon because of the rarity of its use in text strings and its clear visibility. However, most popularity has seemingly been gained by Comma-Separated Values (`*.CSV`) files, and this is what we are going to use here. Depending on the language setting of your computer, a `*.CSV` file may actually has semi-colons as a delimiter, not commas as the name suggests. The delimiter is a comma on English keyboard settings (both American and British), but it is a semi-colon on

several continental European keyboards, depending how they write the decimal "point" which is a "point" in Anglophone countries, but often a comma in continental Europe. Thus to prevent ambiguity, `*.CSV` is not consistently "comma-separated".

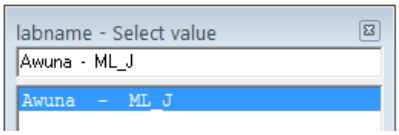
Example: the list with names and code comes as a spreadsheet

We include a spreadsheet formatted to Excel 2003 standard (`a_ex05_namecode.xls`):

	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

Notice here that column A contains the laboratory code (`LABCODE`) and column B contains the laboratory name (`LABNAME`).

The first thing we have to think about is how our label should look in the end. If we remember our current label block as we had it in Exercise 4:

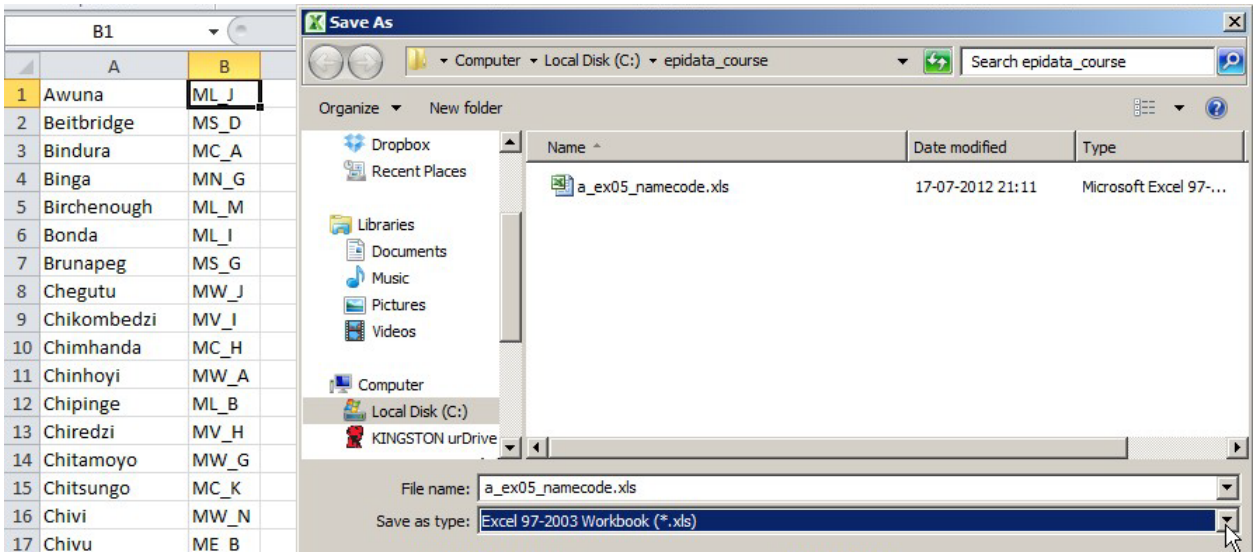


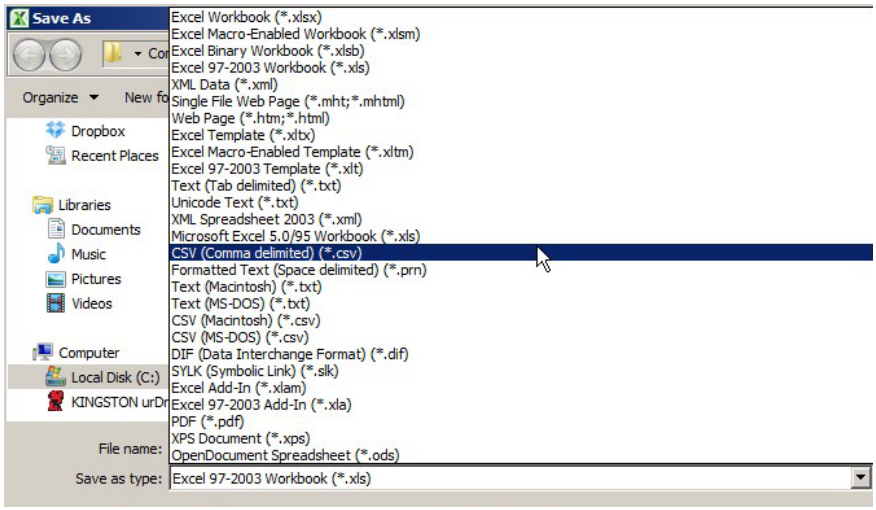
then we see that it was actually the other way around than the way we have it in the spreadsheet: the laboratory name is on the left and the laboratory code is on the right. We need thus to rearrange the columns in the spreadsheet accordingly to become:

	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	Chegutu	MW_J

Then we "Save as" (depending on the spreadsheet software, Excel™ or LibreOffice Calc®) you may need to "Export") "CSV Comma delimited (*.csv)" file.

Save the file as "`a_ex05_namecode.csv`".





By all means, do not trust your spreadsheet to do it right: it is mandatory to check in your text editor software that you really got what you wanted:

a_ex05_namecode.csv

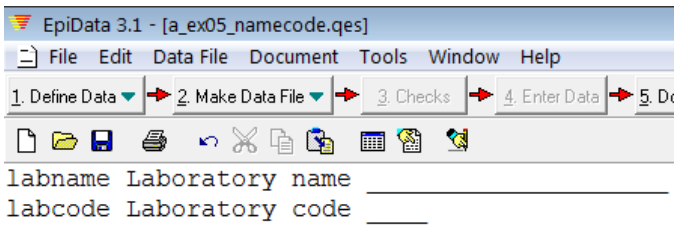
```
1 Awuna,ML_J
2 Beitbridge,MS_D
3 Bindura,MC_A
4 Binga,MN_G
5 Birchenough,ML_M
6 Bonda,ML_I
7 Brunapeg,MS_G
8 Chegutu,MW_J
```

Creating a "a_ex05_namecode.qes" file to provide the structure into which to import the text file

You have learned in the previous exercises that QES files provide the structure for EpiData files. We thus need a QES file into which the text file data can be read. This QES file will have only two fields:

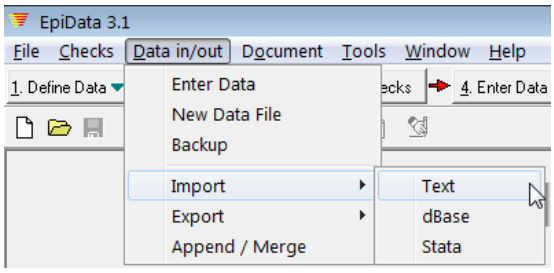
```
labname
labcode
```

and in this sequence as everything has to be in this sequence: Field value then Value label. We have already previously defined that a field length of 20 for LABNAME suffices, while the length for LABCODE is 4, thus we make "a_ex05_namecode.qes" (same name, different extension):

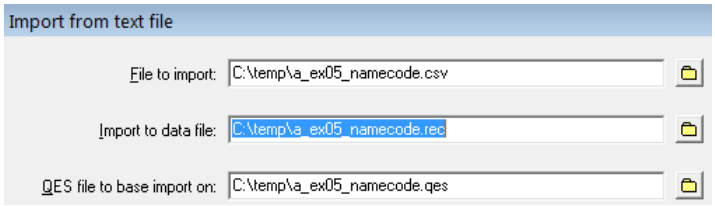


We don't need to make an "a_ex05_namecode.rec" file because the importing of the "a_ex05_namecode.csv" file into the "a_ex05_namecode.qes" file will actually create the "a_ex05_namecode.rec" file.

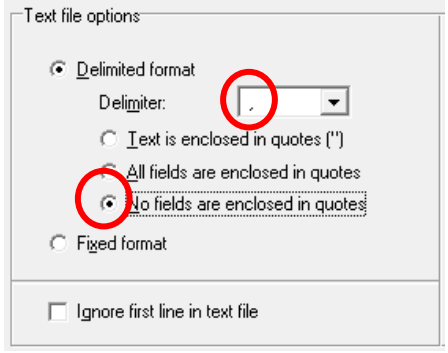
We go straight to:



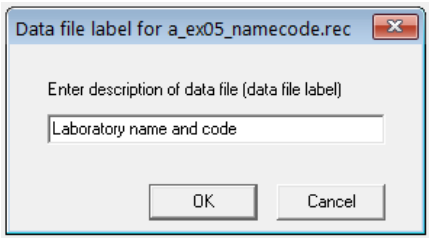
"Data in/out" | "Import" | "Text". If we now pick the "a_ex05_namecode.csv" file, we see why it paid off to use the same file name for the QES file, with the correct naming all falls into place:

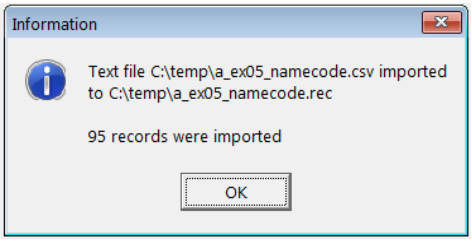


We then carefully check that all relevant boxes are selected / ticked / unticked:



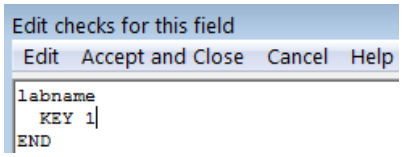
and then we get the *.REC file:





Creating a "a_ex05_namecode.chk" file

We need a *.CHK file. In order to enable the primary file to properly identify the fields in this file as Field value and Value label as the elements for the label block, LABNAME must become KEY 1 and LABCODE must become KEY 2. Add these Checks accordingly:

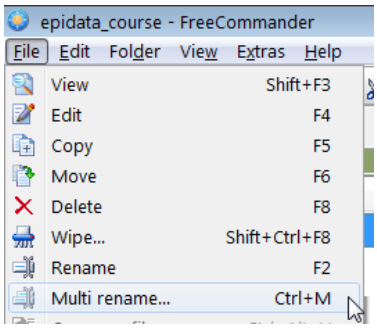


Updating the A_EX04.* to A_EX05.* files

As mentioned earlier there is a bug in EpiData during the export of CHK file. So, if we export an EpiData REC file to another EpiData REC file together with its CHK file, several errors appear in the export of CHK file.

EpiData Entry's option to export data to EpiData format is excellent for a simple (unrelated) EpiData twin set of REC and CHK. It is too buggy for use with a relational database. We therefore do not recommend export.

Alternatively, we create copies of the existing triplet files and rename them appropriately in our file management software (Explorer or FreeCommander) and then do the required edits in all three files. You could use for instance the Multi rename tool in FreeCommander or an analogue in Explorer:



In summary, to make a label block out of an external file, the first pre-requisite is that the external file should be in the REC file format with two fields sequentially arranged to fit into prospective field values and value labels. Also, the two fields should be made key fields - KEY 1 and KEY 2 respectively. Hence, whatever be the format of our original database (Excel, Access, Text), we have to convert that into a REC file. One of the methods of doing it has been explained in this chapter which makes use of the limited options within EpiData

entry. However, there are much simpler/faster options of importing data and saving it as a REC file using EpiData Analysis which is discussed later in EpiData Analysis.

Tasks:

- ***Finalize all to make an update*** A_EX05.QES / REC / CHK ***triplet***
- ***Ensure that the file*** A_EX05_NAMECODE.REC ***serves properly as the new label block***