Exercise 4: Adding field labels and value labels

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

a. Add a Field label to a Field name in the questionnaire
b. Edit the CHK file to show the Value and Value labels
c. Ensuring that the identifier is unique

Field labels and value labels

Our questionnaire shows now the short Field name and the box for the Field value. This is perhaps sufficient if we enter the data ourselves but it might be difficult for a data entry person not familiar with the definition of the fields. EpiData Entry allows the addition of Field labels to the Field name which become part of the Field definition. As the Field values are numeric, i.e., for instance "1" to indicate "Female sex", the data entry person does not see after entry whether the value that was entered was really the value intended. It is possible to show the Value label to the left of the Value. Adding these two components to the QES (the Field label) and the CHK file (the instruction to show the Value label) is part of this exercise. In summary, what this exercise is to accomplish is to have the following display shown here for the Field SEX:

Field name	Field label	Value	Value label
sex	Sex of examinee	9	No sex recorded

The Field label is added to the questionnaire, while the instruction to display the Value label is added to the Check file with TYPE COMMENT as follows:

```
sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
  TYPE COMMENT
END
```

Ensuring that the identifier is unique

The field SERNO will later be used as a ***unique identifier***. That it is, is relatively easily to ascertain in the particular case of the laboratory register where the serial number is entered sequentially. But it would be impossible to know whether a particular identifier had not been used before if identifiers are codes of the type AX7, ZV4, YY3, etc, and in no particular sequence. It is possible to tell EpiData Entry to check whenever an identifier is entered whether it has ever been used before and is thus not unique. The command that needs to be added is:

```
serno
  MUSTENTER
  KEY UNIQUE
END
```

or

```
serno
  MUSTENTER
  KEY UNIQUE 1
END
```

The command KEY in a given field results in the writing of another file with the extension *.EIX, a so-called index file. Index files allow a very rapid search for specific information. Thus if the field SERNO is a KEY UNIQUE field, every time a serial number is entered, the entire database is searched whether this registration number is truly unique. If it is not, a warning will inform the data entry person that the identifier had been used before and in which record and whether the person wants to see that record. It will be impossible to continue to enter data without entering a truly unique identifier.

The number 1 in KEY UNIQUE 1 above is optional. If not used, internally KEYs are numbered sequentially up to the maximum allowable 10 KEY fields that can be used in a record.

If a data entry person uses the mouse to bypass the field SERNO then the command KEY UNIQUE cannot be executed and it is possible to save the record to disk without the identifier. This will be possible without any problem for 1 record, but will pose a problem during validation which will be on the KEY. If an attempt is made to save a second record without SERNO, the data entry person will be warned that the KEY is not unique. This is so because the missing value is represented by a period (.) and having twice a period will make it non-unique. While it is bad to have missing values in MUSTENTER fields (indicating that the mouse was used), it very troublesome not to have a unique identifier.

The best strategy is thus to prevent the possibility to save a record without an identifier. We add thus to the CHK file an AFTER RECORD command that checks whether the identifier is present, to issue a warning is not, and get the data entry person back to the identifier field:

```
AFTER RECORD
  IF serno=. THEN
   HELP "Core information missing:\n SERNO\n must be available" TYPE=WARNING
   GOTO serno
  ENDIF
END
```

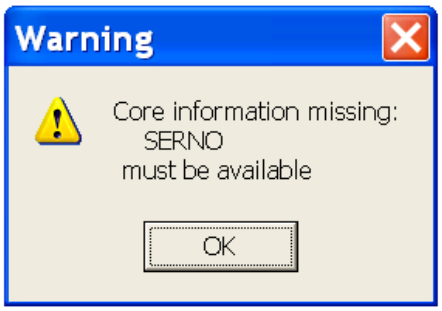
The command AFTER RECORD concluded by the command END contains nested commands that are executed before the record can be saved. It checks whether the field SERNO has missing information (“.”), and if so will take the data entry person back to the field SERNO with the following:

```
  IF serno=. THEN
   GOTO serno
  ENDIF
```

In order to inform the data entry person why the cursor will go back to the field SERNO, the command HELP is added. As it is written above, the line:

```
   HELP "Core information missing:\n SERNO\n must be available" TYPE=WARNING
```

Will result in the display of a WARNING box:



Task

- ***Revise the A_EX03.QES file, save it as A_EX04.QES, make the REC file, open the A_EX03.CHK file, save it as A_EX04.CHK and make the necessary changes and then enter some made-up data to check the functionality.***