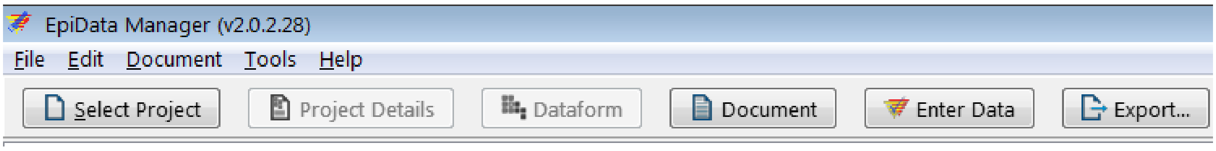
Exercise 2: Create a basic data entry form

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

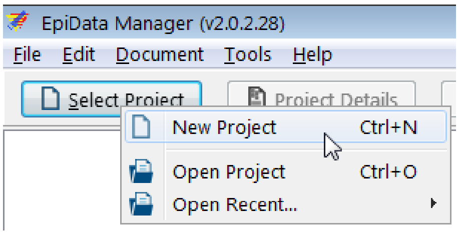
a. Create and edit a data entry form with EpiData Manager.
b. Include study information and password protection.
c. Use headings and sections.
d. Create basic label blocks, understand “Must enter”, jumps, ranges, using label blocks for missing / out of range numeric data.
e. Align properly horizontally and vertically both for aesthetics and functionality.

You are now ready to start with the design of the questionnaire with EpiData Manager, based on the data documentation sheet prepared in Exercise 1.

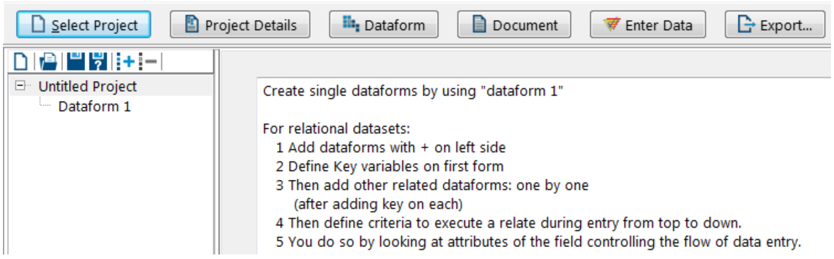
When we open EpiData Manager, we have the interface with three rows, displaying the version in the top row, menu items in the middle row and some specific menu items in the bottom row:



As we start from scratch, we begin with Select Project | New Project (or shortcut **CTRL+N**):



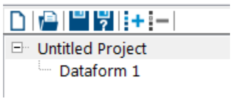
We note the opening of the set up:



Create single dataforms by using "dataform 1"

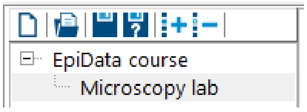
For relational datasets:
1 Add dataforms with + on left side
2 Define Key variables on first form
3 Then add other related dataforms: one by one
(after adding key on each)
4 Then define criteria to execute a relate during entry from top to down.
5 You do so by looking at attributes of the field controlling the flow of data entry.

The information on the right side informs about the differences of using a single data entry form (as we will do now) or creating a relational dataset (as we will do in the last exercise of Part A). Hovering over the small icons:

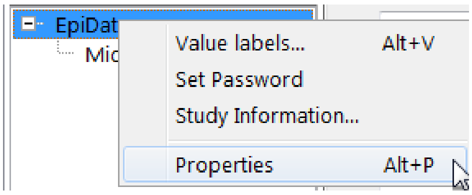


shows what they accomplish.

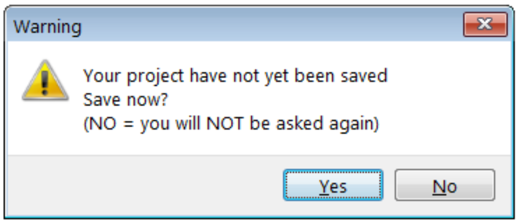
Clicking on Untitled Project and pressing function key F2 allows us changing the name to, say, “EpiData course”, and clicking on the Dataform 1 and pressing function key F2 allows us to change the name of the form to “Microscopy lab”, so that we will get:



Right-clicking on the Project name gives us options:



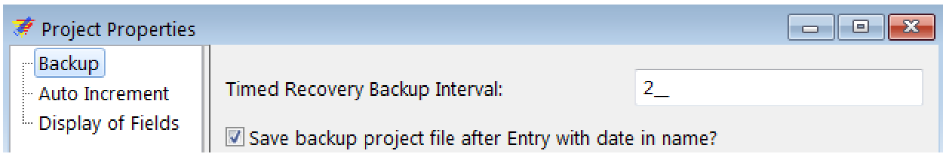
As we play around, at some point in time, EpiData Manager will prompt us to save the project:



Your project have not yet been saved
Save now?
(NO = you will NOT be asked again)

And we will do so promptly, saving the project as **a_ex02.epx** (the extension is supplied and the suggested folder is our chosen data folder C:\EPIDATA_COURSE).

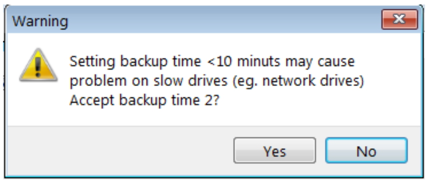
To resume from the above project options, let’s check Properties (shortcut **ALT+P**). You note that we get a window with something we had already defined earlier during set-up in Preferences:



Timed Recovery Backup Interval: 2_

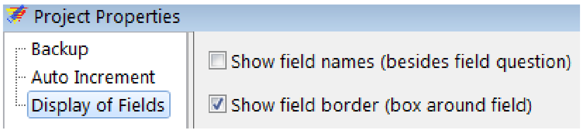
☑ Save backup project file after Entry with date in name?

If we click on any of the other two, e.g. Display of Fields, we get a small warning (also mentioned during set-up):



Setting backup time <10 minuts may cause problem on slow drives (eg. network drives)
Accept backup time 2?

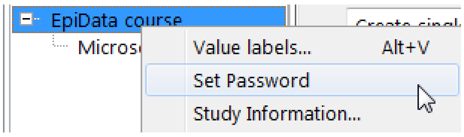
We accept and get what we had already accepted as default in Preferences:

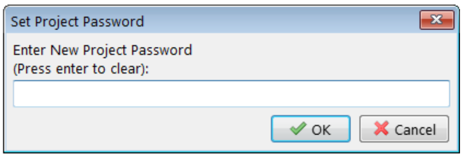


□ Show field names (besides field question)

☑ Show field border (box around field)

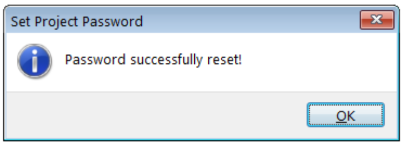
There are thus different possibilities to set our preferences and to change them here at any time without first going back to the initial settings. Thus we OK and check out Set Password with a right-click:




Set Project Password

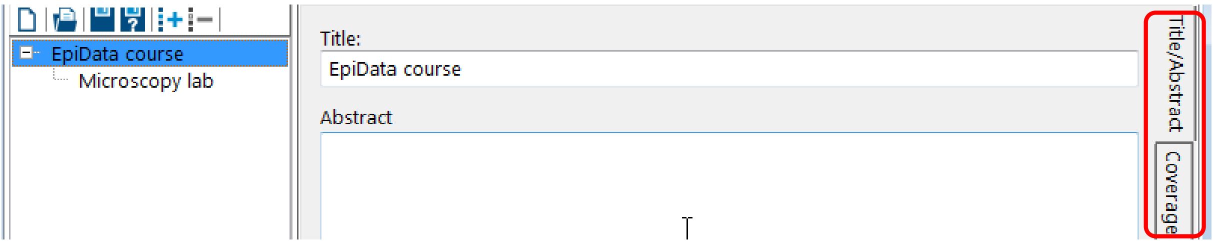
Enter New Project Password
(Press enter to clear):

There are different levels of password setting, this is for the entire project, but a password can also be set for sections within a project. For the time being, we do not wish to set a password for the project and we clear with the Enter key and get:


Set Project Password

Password successfully reset!

Do you note the difference between when we click on the project name or the data form name respectively? Clicking on the project name "EpiData course" you get:



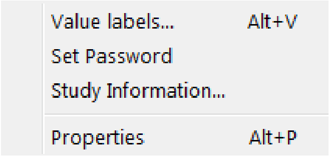
Title: EpiData course

Abstract

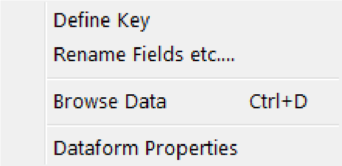
At the very right (circled red) you see that this is the first of the several vertically aligned tabs, further including Coverage, Description, Ownership, Funding, and Version Details. In the current tab we could add an Abstract, clicking on the second tab, Coverage, we can add information on Geographical location, Language, Data time coverage, Population, and Unit of observation. The third tab, Description, allows entering Keywords, Purpose, Citations, and Design. The forth tab, Ownership, allows entering Organisation/Institute, Agency, Authors/Contributors, and Rights. A fifth tab, Funding, is a free text field, and the final tab, Version, permits entering information about the study and the study form. Short, this menu option allows thorough documentation of the study. It is highly recommended that this is actually used to its fullest for a real study. For the purpose of this course, we will leave it unfilled.

Contrasting right-clicking the Project name and the Dataform name we get:

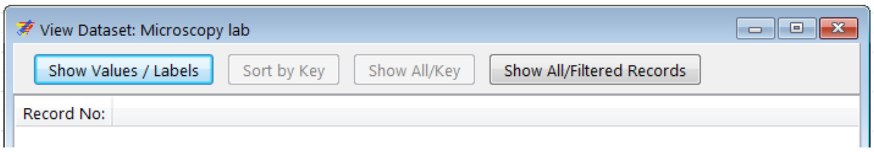
Project name: EpiData course



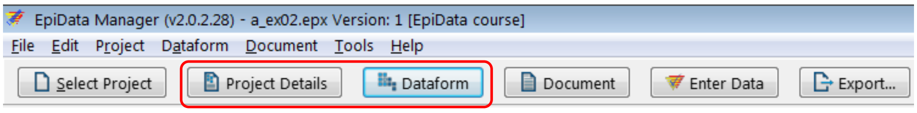
Dataform: Microscopy lab



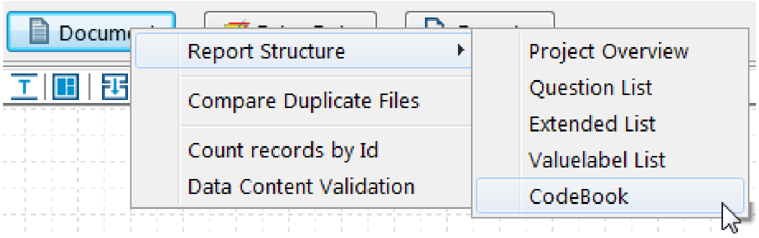
We have already checked out some of these options, while others will be used in what is coming once we start to actually make the data entry form, while further ones like Browse Data will only become available once we have data, while for now, as expected, an empty window opens:


View Dataset: Microscopy lab

The two sub-menus above can also be accessed any time from the Project Details and Dataform menus respectively:


EpiData Manager (v2.0.2.28) - a_ex02.epx Version: 1 [EpiData course]
File Edit Project Dataform Document Tools Help
Select Project
Project Details
Dataform
Document
Enter Data
Export...

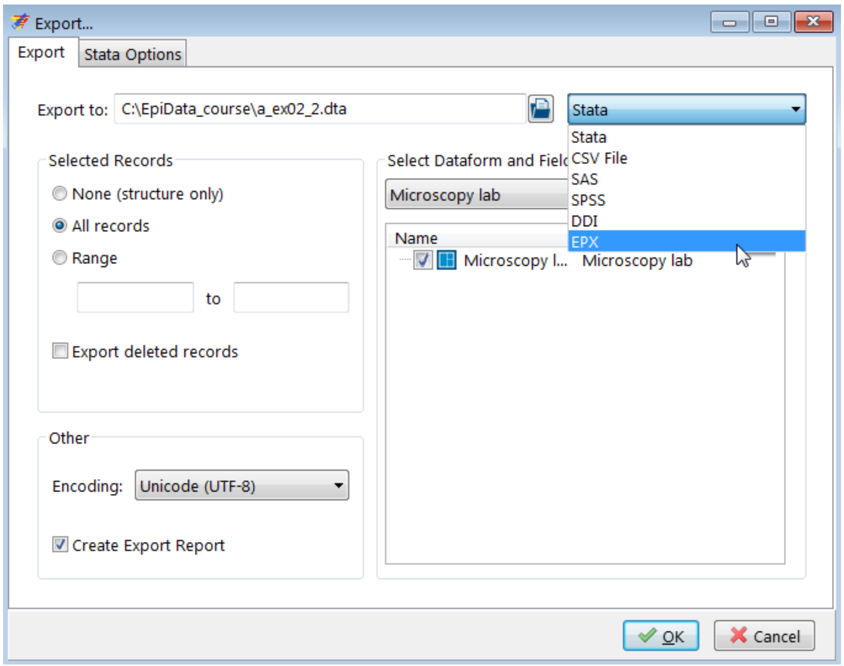
The third menu point above, Document, gives a sub-menu and the sub-menu item Report Structure has itself a sub-menu:



some of which we will be using at a later point of the course.

If the EpiData EntryClient software module is located in the same folder as the EpiData Manager as is the case in our set-up (and it is recommended not to keep them separately), we have direct access to it from the second-to-last menu item Enter Data:

The last menu item, Export, gives multiple options to export the structure or the data to various formats:



Export...

Export | Stata Options

Export to: C:\EpiData_course\a_ex02_2.dta

Stata

- Stata
- CSV File
- SAS
- SPSS
- DDI
- EPX

Selected Records

○ None (structure only)

● All records

○ Range ____ to ____

□ Export deleted records

Select Dataform and Field

Microscopy lab

Name	
☑ Microscopy l...	Microscopy lab

Other

Encoding: Unicode (UTF-8)

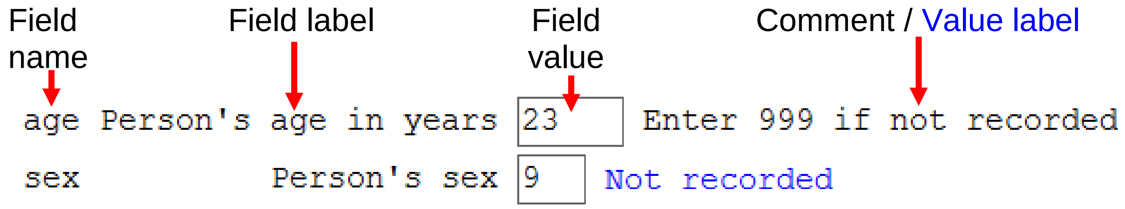
☑ Create Export Report

OK Cancel

We will be making use of this menu item at a later point in time.

Designing the data entry form

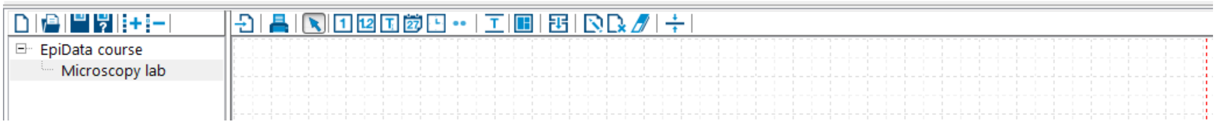
Before starting with the actual design of our data entry form, let’s clarify the terminology and where what is going to be placed. There are four elements that go with a field:



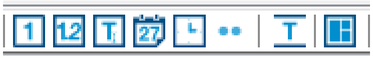
```
age Person's age in years 23  Enter 999 if not recorded
sex             Person's sex 9 Not recorded
```

We call the variable the **Field name** (see Exercise 1). As shown in Exercise 1, the Field name may or may not be displayed. In setting up Preferences, we opted for not displaying it. To have more explicit information than what can be conveyed by the length 10, single-word Field name, we supplement the Field name with a more explanatory **Field label**. The third element on the line is the Field definition which receives the actual **Field value** during data entry. The Field value is the actual datum (singular of "data") for that variable for that observation. For *continuous variables*, we might add an explanatory **Comment** to be displayed in a Note during data entry, here for instance what to do if the primary source has no information on this variable for this observation. For *categorical variables*, the comment is not necessary: a label block will comprehensively provide information on what is entered. As we use numeric coding, it would, however, be a nice touch if after picking a value from a pick list (label block), the **Value label** would appear to the right of the field, as a kind of visual confirmation that what had been intended was actually picked.

If we click in the Dataform tab "Microscopy lab", we look now at the data form canvas:



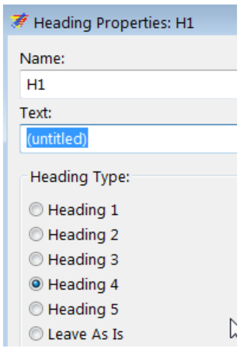
Above the grid we see various icons, most importantly right now those that symbolize different variable types which we can identify by hovering with the cursor over each one:



From the excerpt here of these icons, from left to right, these are:

New Integer Field
New Float Field
New String Field
New DMY Field
New Time Field
Other Field Types…
New Heading
New Section

We start with the second-to-last on the left, the New Heading, click on it and get the menu:



Heading Properties: H1

Name:
H1

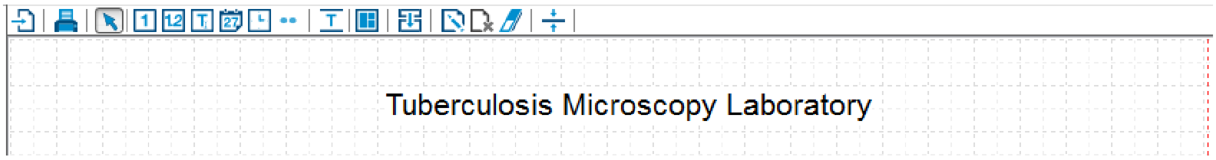
Text:
(untitled)

Heading Type:
○ Heading 1
○ Heading 2
○ Heading 3
● Heading 4
○ Heading 5
○ Leave As Is

We overwrite the Text with:

Tuberculosis Microscopy Laboratory

Those with some notion of HTML (the language of the internet) may recognize that text size is expressed relatively, where Heading 1 is the relatively largest and Heading 5 the relatively smallest. Let’s choose (arbitrarily) Heading 2 and click Apply to get our heading:

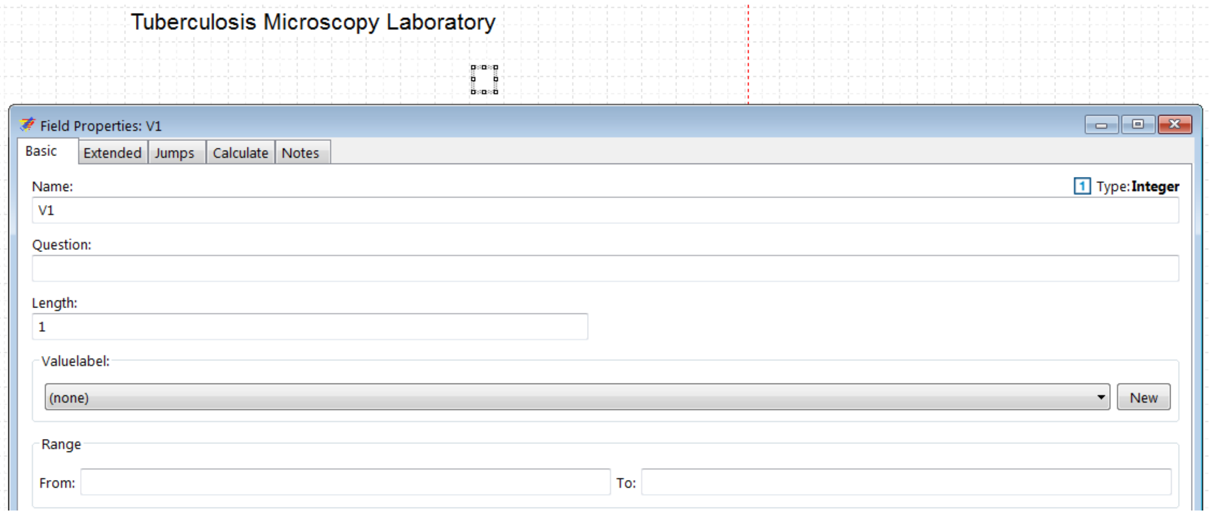


Tuberculosis Microscopy Laboratory

You can drag this text box around to your preferred position. By right-clicking it, you can Edit it at any time to change the wording or the size if you wish to do so.

In some larger data entry forms, we then may next choose to make sections to which we could give different attributes, but for the time being, we will simply start with our fields.

Consulting the Data Documentation Sheet, we see the first field to be `serno`, the `Laboratory serial number`, an integer field of length 4. Clicking on the icon for the integer field, then clicking into a suitable place on the canvas, we get the Field Properties menu:



Tuberculosis Microscopy Laboratory

Field Properties: V1

Name: V1

Type: Integer

Question:

Length: 1

Valuelabel: (none)

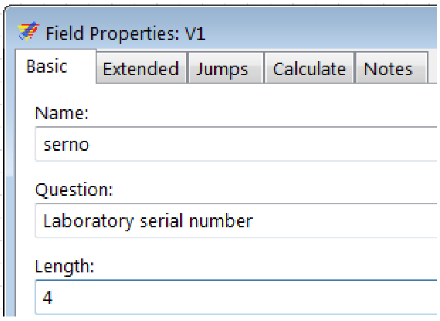
Range

From:

To:

This is our main working tool for the design of the data entry form. We will therefore complete each page where applicable and study carefully the options on each tab.

The field Name is given the default name V1 and we change this to `serno` as defined in the Data documentation sheet. The Question is for the Field label which we defined as `Laboratory serial number`, and its Length is 4. It has no Valuelabel (it is not a categorical variable) and we have no Range to assign, thus completed we get:



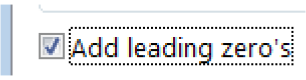
Field Properties: V1

Name: serno

Question: Laboratory serial number

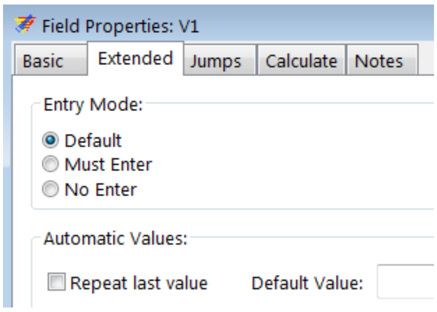
Length: 4

At the bottom of the Basic tab we tick to add leading zero's:



☑ Add leading zero's

We switch to the tab Extended showing (at the top):



Field Properties: V1

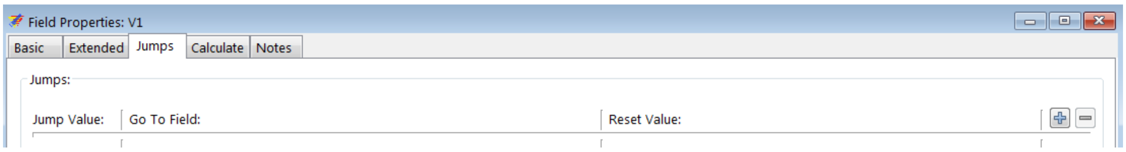
Entry Mode:

● Default
○ Must Enter
○ No Enter

Automatic Values:

□ Repeat last value   Default Value:

The Entry Mode is of critical importance and there are three possibilities. The Default is "you do as you wish", i.e. the data entry person is allowed to enter or not to enter a value into the field. The second option Must Enter signifies that a value has to be entered else the data entry person cannot continue to work (or if bypassing the field by using the mouse, will be stopped latest when trying to save the record). This is our preferred method because "forgetting to enter" is not the same as "there is no information" and by forcing Must Enter a value must be entered. Therefore, whenever a value for a variable is missing, we must tell the data entry person what has to be entered in such a situation. We will thus make every field a Must Enter field with two exceptions. The first exception is for calculated fields, something we will use in a later exercise when we create a composite identifier. The second exception is for fields that can be bypassed with Jumps as we will see shortly. For the `Laboratory serial number` we thus tick Must Enter. There is a possibility for Automatic values, such as Repeat last value. If we tick the latter, the next new record will inherit the value from the current one for this field. This is convenient for instance if a long sequential series of records always has the same date before it changes. Whenever we change the inherited value by overwriting it in a new record, the new value will be passed on as a repeat value to the next record. Obviously, this is not what we need for the `Laboratory serial number`. After having ticked Must Enter, we thus move to the tab Jumps:



Field Properties: V1

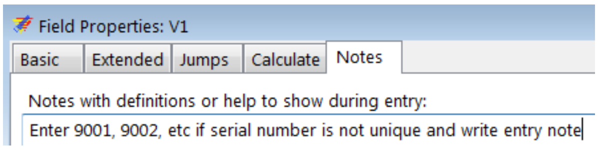
Jumps:

Jump Value:	Go To Field:	Reset Value:

Jumps are used to define whether based on a value in the current field the next field is to be bypassed or not. If the field asks for smoking status and the response is non-smoker then the next field that would quantify the number of cigarettes per day is bypassed while it is not if the response in the current field is smoker. We will use this with the results fields.

The Calculate tab allows creating the value for a new variable from the values of one or more existing variables. For instance one could calculate the value of a date field from three date components. We generally discourage the use of calculations during data entry, they belong to the analysis, with one important exception though. Sometimes, a unique identifier for a record can only be defined by constructing from more than one variable. We will show this in a later exercise.

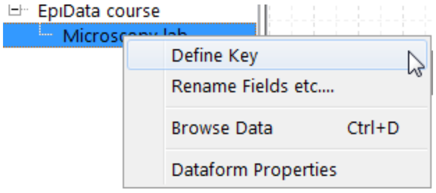
What we write into the Notes tab will briefly pop up if the cursor is in the field. For the Laboratory serial number we actually foresee such a situation, and this is to instruct to enter “artificial” serial numbers 9001, 9002, etc. if we happen to stumble upon a non-unique serial number. We thus enter a Note as for instance:



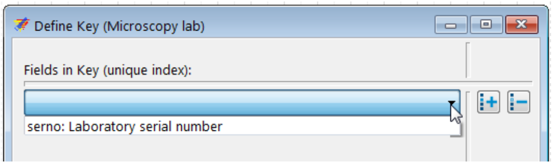
Notes with definitions or help to show during entry:

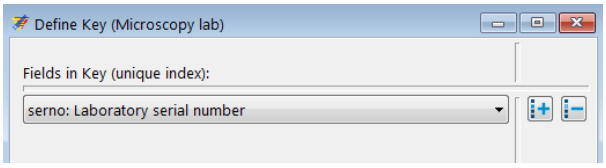
Enter 9001, 9002, etc if serial number is not unique and write entry note

As an interlude here, we need to clarify key issues about the identifier. No variable is of more importance in a record than the unique identifier. We can even state that a record is useless without a unique identifier. For instance, if we have no value for age in a record, we can always check whether it is truly missing from the original case report form if we can identify that case report form by the unique identifier. Without it, we cannot possibly learn the truth. It is impossible to validate our data with double-entry if we do not have unique identifiers. For persons not familiar with the tuberculosis microscopy laboratory register, this is what you need to know: each examinee who presents to the laboratory for a microscopic examination is given a serial number which starts with 1 each calendar year. An examinee might thus provide a series of up to 3 specimens and the examinee (not the examination) is the unit of observation and “numbered” as such. The Laboratory serial number is unique for an examinee presenting him- or herself newly for a serial set of examinations usually over a one- to three-day period during the course of 1 year in 1 laboratory. We have not yet instructed EpiData Manager to treat the Laboratory serial number as a unique identifier. This is not done here in the Field Properties, it has to be done for the file (you may remember the options by right-clicking the data form name Microscopy lab):



It is of critical importance to Define Key, thus when done, we close the Field Properties dialog menu and go to the data form Microscopy lab, click on Define Key and select it:


Define Key (Microscopy lab)
Fields in Key (unique index):
serno: Laboratory serial number

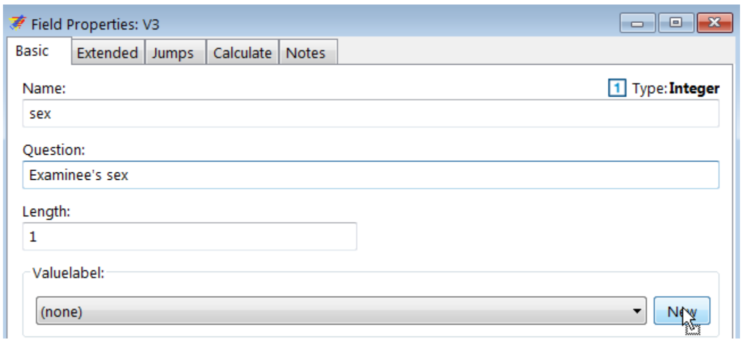


serno: Laboratory serial number

What this will do is that whenever you enter a Laboratory serial number in a new record, EpiData will check the entire data set whether or not this identifier has already been used. If not, you can continue your work. If it had been used in another record, the EpiData EntryClient will tell you in which record and offer to go to it to change it if it had been entered erroneously or to enter an alternate in the current one (as suggested here in Notes). In any case, you cannot continue working unless the identifier is assuredly unique for the file.

Our next field is the Registration date with the field name regdate. We thus pick the icon for New DMY Field and start to complete the Basic tab of Field Properties. We did define a legal range. Enter it and make the field Must Enter.

The third field is sex which we defined as integer for length 1 and we need to create here a new Valuelabel:



Name: sex

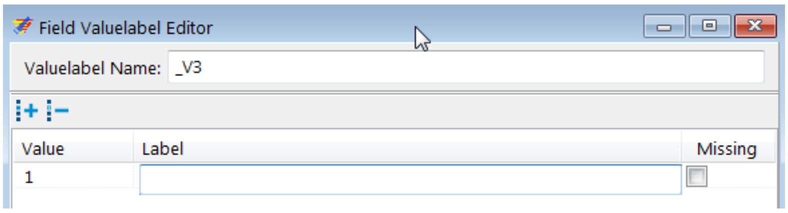
Type: Integer

Question: Examinee's sex

Length: 1

Valuelabel: (none)

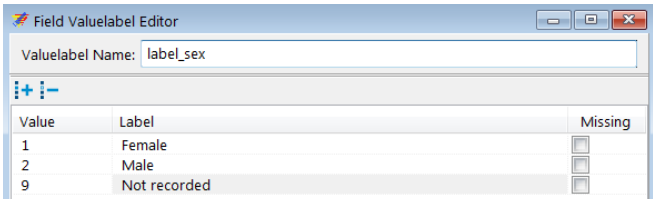
The sub-menu for the Valuelabel:



Valuelabel Name: _V3

Value	Label	Missing
1		□

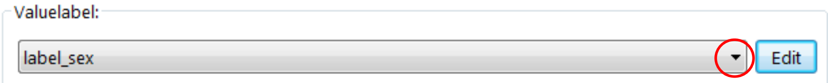
It is quickly completed – note that we give a sensible, easily identifiable Valuelabel Name. Though we are quite free how we do that, taking the default is not very informative, thus:



Valuelabel Name: label_sex

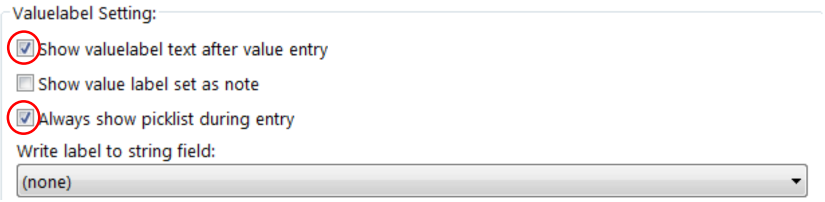
Value	Label	Missing
1	Female	□
2	Male	□
9	Not recorded	□

We could tick the value for Not recorded as Missing, it might or might not come handy later in the analysis, but we leave it unticked for the time being. After accepting, the Valuelabel Name will get inserted automatically as Labelvalue into the foreseen box in the Basic tab:



Valuelabel: label_sex

else we can pick it from the drop-down list (red circled here). In the Extended tab we make the field Must Enter and tick the two boxes applicable for the Valuelabel Setting:



Valuelabel Setting:

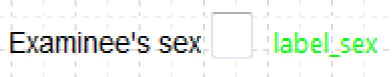
☑ Show valuelabel text after value entry

□ Show value label set as note

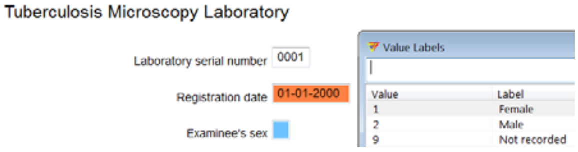
☑ Always show picklist during entry

Write label to string field: (none)

The first (Show valuelabel …) is ticked by default, but the Always show picklist … must be set. What the two accomplish is that the value goes into the field box and the Valuelabel is written to the right of the field box (for visual verification), but before this is done, the Picklist with the Value-Valuelabel pairs is popping up ready for selection as soon as the cursor gets into the field. In EpiData Manager, it gets indicated in green that we have a given label block:



In the EntryClient, ticking both boxes will give the pop-up (left) and the writing of the value label (right) after the selection of the value:



Tuberculosis Microscopy Laboratory

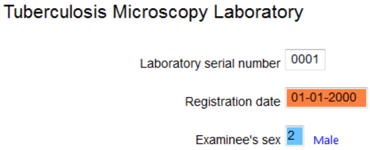
Laboratory serial number 0001

Registration date 01-01-2000

Examinee's sex

Value Labels

Value	Label
1	Female
2	Male
9	Not recorded



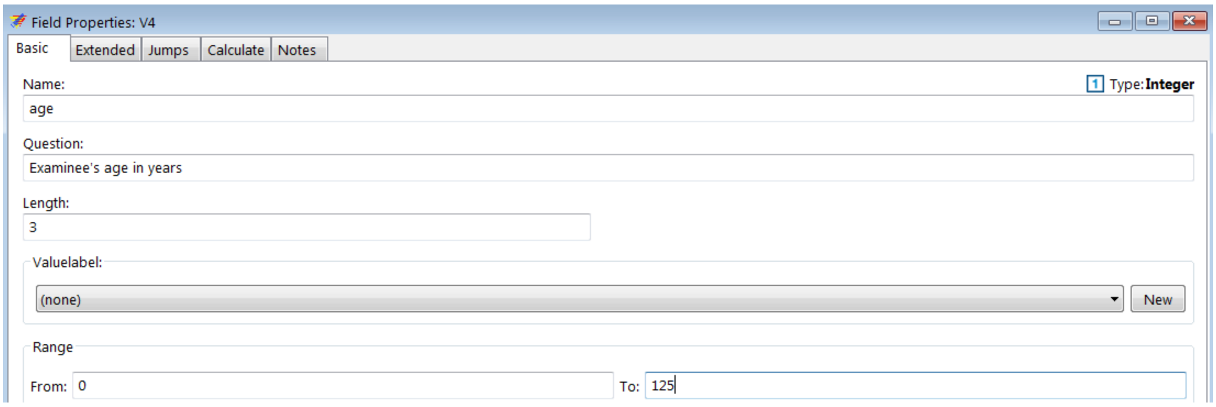
Tuberculosis Microscopy Laboratory

Laboratory serial number 0001

Registration date 01-01-2000

Examinee's sex 2 Male

The fourth variable is `age`, the `Examinee's age in years`, with a field length of `3`, and for which we allow a range from `0` to `125`, and a legal value of `999` for missing `age`:



Field Properties: V4

Name: age

Type: **Integer**

Question: Examinee's age in years

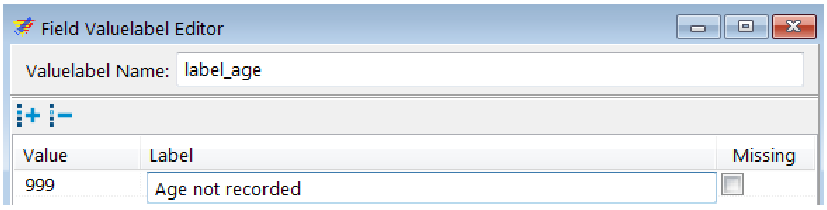
Length: 3

Valuelabel: (none)

Range

From: 0 To: 125

The out-of-range legal value is put into a Valuelabel, thus pick New and add as follows:

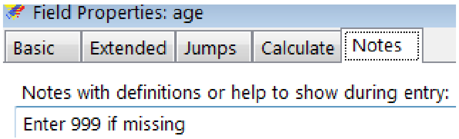


Field Valuelabel Editor

Valuelabel Name: label_age

Value	Label	Missing
999	Age not recorded	□

Then set to Must Enter in the Extended tab, but set / keep all boxes in Valuelabel Setting unticked: the value-label pair is just to “legalize” the value `999`, not to have a rudimentary pop-up window. Instead, we instruct the data entry person in the Notes tab what to do if the value for `age` is missing:



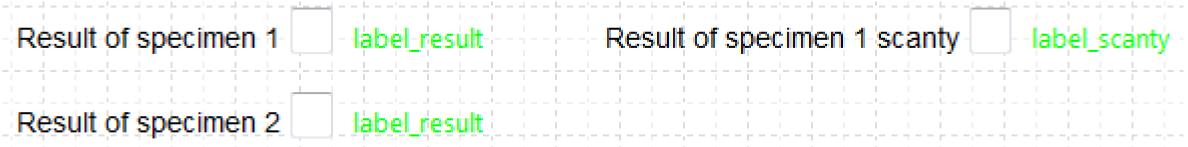
Field Properties: age

Notes with definitions or help to show during entry:

Enter 999 if missing

The fifth variable is `reason`, the `Reason for examination`, an integer of length `1`. The principle is the same as for `sex` and nothing is new here above what was learned before.

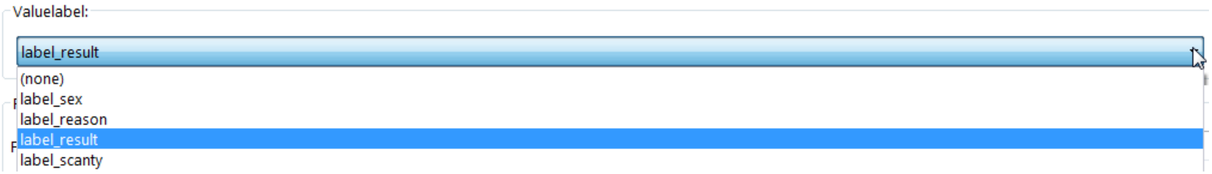
The sixth variable is `res1`, the `Result of specimen 1`, an integer of length `1`. Again, the principle is the same as for the other categorical variables with label blocks, but we do have something new. We will insert a Jumps command for every value, except if it is `6`, `Scanty quantified`. Because we need to tell EpiData where to go, if the field has or has not the value `6`, we propose that everything else but the `Jumps` command for fields `res1`, `res1sc`, and `res2` is completed first, and then we return to `res1` to formulate the Jumps command (not critical here as will be seen). We don’t have to, but let’s arrange the fields for `res1` and `res1c` (and the other “pairs”) horizontally:



Result of specimen 1 □ label_result Result of specimen 1 scanty □ label_scanty

Result of specimen 2 □ label_result

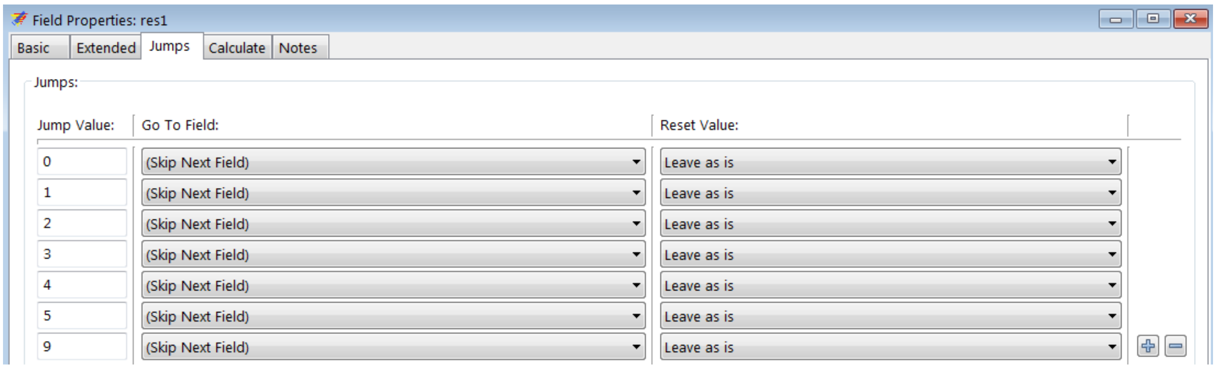
Note also that we do not need to write a label block for each result and for each scanty result, it suffices to write a generic label block for results as all three results have the same Valuelabel blocks and we may name it `label_result` and name the Valuelabel block for scanty results something like `label_scanty`. After we have it for the first, we can pick it from the drop-down list for the Valuelabel:



Valuelabel:

label_result

Once we are done with these, we return to the field `res1` and open the Jumps tab to enter what we require. Because the field to skip is the next field (`res1sc` here), we do not actually need to state to which field to go. We just add all values except 6 and the instruction (Skip Next Field) under the Go To Field. Click the + sign to the right to add a new line and enter the next value:



Jumps:

Jump Value:	Go To Field:	Reset Value:
0	(Skip Next Field)	Leave as is
1	(Skip Next Field)	Leave as is
2	(Skip Next Field)	Leave as is
3	(Skip Next Field)	Leave as is
4	(Skip Next Field)	Leave as is
5	(Skip Next Field)	Leave as is
9	(Skip Next Field)	Leave as is

Tasks:

- ***Finalize the data entry form for the remaining results variables. Note that the fields for scanty results cannot be Must Enter. Note further the options in the Drop down menu for what is the default “Skip Next Field” to pick the best option for the last result.***
- ***Align the field correctly using the Alignment icon. Note that correct vertical alignment is critical if two variables are on the same horizontal pane (EpiData Manager gives nice blue and red guiding lines). If two variables on the same horizontal pane are vertically mismatched, the sequence of data entry will go wrong!***
- ***Tell EpiData Manager to create the Codebook and save the output file as a text file “a_ex02_codebook.txt”.***