

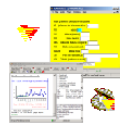
Exercise 4: Creating a menu for standard reports

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

- a. Write an HTML-based interface for a menu
- b. Writing programs with interactive prompting

This exercise involves working with HTML to make a user-friendly interface with menus and to prepare the EpiData Analysis programs that execute the production of standard reports with interactive prompting for choices.

The end product interface will appear as follows:



Quarterly reports on case finding and treatment outcome



Enter data



Access EpiData Entry

Create reports



A quarterly report on case-finding



A quarterly report on treatment outcome

Terminate session



Erase temporary session files



Exit the report menu

We will structure the exercise as follows:

Preparatory work

Installing EpiData Entry and EpiData Analysis

Delete an obsolete sub-folder and create new sub-folders

Unzip the required files from the course website into the relevant sub-folder

Background how EpiData Analysis starts and how to shape its looks when opening

Making the interface in HTML

Using an HTML editor to make the skeleton of the interface

Editing the HTML file

Making the EpiData Analysis programs

The program to produce the quarterly report on case finding

Make the basic dataset

Make the interactive selection process

Make the charts side by side
The program to produce the quarterly report on treatment outcome
Make the basic dataset
Make the interactive selection process

Preparatory work

Installing EpiData Entry and EpiData Analysis

First make sure your “Normal EpiData Analysis” is updated to the newest version. The recommended location for both EpiData Entry and EpiData Analysis is C:\EPIDATA.

In addition, we will install EpiData Entry and EpiData Analysis into a separate folder (make it user-defined to keep control over what’s going to happen). When prompted for the path, put it into:

```
c:\epidata_report
```

You will get in this folder three sub-folders and 24 files:

```
docs\  
samples\  
TestData\  
English.ea.lang.txt  
Epdintro.pdf  
EPIDATA.CNT  
EpiData.exe  
EPIDATA.HLP  
epidata.ini  
EpiData.lbl  
epidatastat.10  
epidatastat.12  
epidatastat.15  
epidatastat.20  
EpiDataStat.exe  
epidatastat.ini  
epiout.css  
epiout_b.css  
epiout_w.css  
license.txt  
licensetest.txt  
preventdouble.ea  
readme.rtf  
unins000.dat  
unins000.exe  
unins001.dat  
unins001.exe
```

Delete unnecessary sub-folders and create new sub-folders

The TestData and samples sub-folders are superfluous for this exercise and can be deleted. Instead create **six new** sub-folders, so that you have a total of seven sub-folders:

```
data\  
docs\  

```

images\
originals
pgm\
required\
temp\
English.ea.lang.txt
...

Unzip the required files from the course website into the relevant sub-folder

The course website contains a zip file with 26 required files:

choice_year.qes
choice_year.rec
epidata.png
epidata.ini
epidata_wikiL.png
epidatastat.ini
epidatastatsetup.ini
epidatastat.png
epiout_monospace.css
eraser.png
insert_pgm_html.txt
next.gif
quit.jpg
sample_2004.chk
sample_2004.eix
sample_2004.qes
sample_2004.rec
sample_2005.chk
sample_2005.eix
sample_2005.qes
sample_2005.rec
sample_2006.chk
sample_2006.qes
sample_2006.rec
side_by_side_graphs.html
start_template.htm
union.jpg

Unzip all these 27 files into the \REQUIRED sub-folder. Then move (or copy) and over-write if necessary as follows:

From the \REQUIRED sub-folder to the \IMAGES sub-folder:

epidata.png
epidata_wikiL.png
epidatastat.png
eraser.png
next.gif
quit.jpg
union.jpg

From the \REQUIRED sub-folder to the \ORIGINALS sub-folder:

choice_year.qes
choice_year.rec
sample_2004.chk

```
sample_2004.eix
sample_2004.qes
sample_2004.rec
sample_2005.chk
sample_2005.eix
sample_2005.qes
sample_2005.rec
sample_2006.chk
sample_2006.qes
sample_2006.rec
side_by_side_graphs.html
```

From the \REQUIRED sub-folder to the \EPIDATA_REPORT root folder:

```
Epidata.ini
epidatastat.ini
epidatastatsetup.ini
```

Note to the above three *.ini files: you could perfectly well edit all these three files (and we show below how to edit the epidatastat.ini to serve our purpose in some detail).

From the \REQUIRED sub-folder to the \DOCS\EN sub-folder:

```
start_template.htm
epiout_monospace.css
```

This should leave you with a single file in the \REQUIRED sub-folder:

```
insert_pgm_html.txt
```

This is an example that we may make use of later when we insert HTML code to call a program.

Background how EpiData Analysis starts and how to shape its looks when opening

In any software you may use, there is an executable file that starts the process of opening the program and displaying the standard interface. In EpiData Analysis, this file is in the root of the folder in which you installed the program and its name is:

```
EpiDataStat.exe
```

This file contains all the essential code to direct EpiData Analysis to do what it is expected to do. Publishing this code will make EpiData Analysis what we call “open-source” software. While the designers and programmers of EpiData software are working on preparing this source code with sufficiently detailed documentation to ultimately make it open-source, the time is not yet quite mature to do so because the documentation must be un-ambiguous and clear for any other developer to derive usefulness for further development from it. As we are not software developer ourselves, we do not have any need to know the source code, the only thing we need to know is some very basic things that this executable file does, and how we can override certain things that we require to meet our needs.

Among other files, the executable file looks first for an HTML file that is located in the \DOCS\EN sub-folder:

start.htm

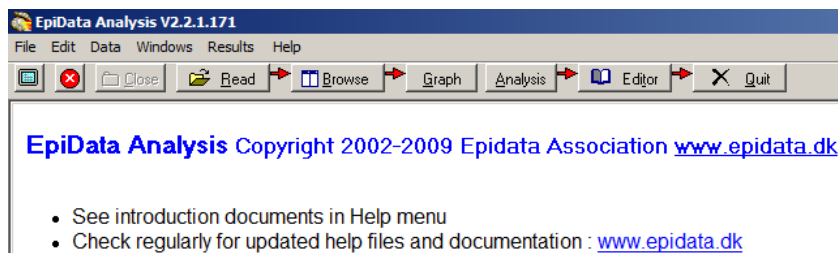
If we double-click this file it opens in our default browser and this is the display we get:

EpiData Analysis Copyright 2002-2009 Epidata Association www.epidata.dk

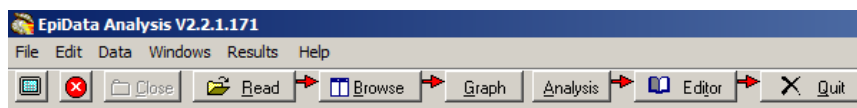
- See introduction documents in Help menu
- Check regularly for updated help files and documentation : www.epidata.dk

We can change this visualized part to our liking in the `start.htm` file which will be one of our tasks.

When we open EpiData Analysis, we see in addition the version display, the menu bar, and the process bar:

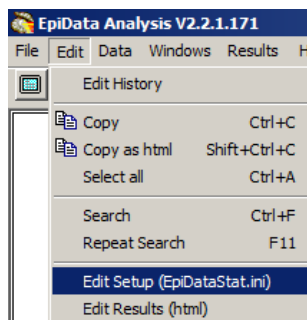


The upper component:



is part of the default defined in the executable file. It is, however, possible to suppress part or all of it by modifying the `EpiDataStat.ini` file.

With the opening of EpiData Analysis, a small program called `EpiDataStat.ini` is run. You can access it from the Edit menu:



If you choose to edit this Setup program, it will open in the Editor. Its initial default script is:

- * EpiData Analysis default settings file
- * Edit the next lines to change size or font
- set echo=off

- * Viewer font and size: (plus editor and help windows)

```

set browser font size =12
set graph font size =12
set viewer font size =12
set window font size =12
set editor font size =12
set viewer font name ="Verdana,Courier"

* uncomment next two lines to display Chinese Characters
*SET viewer font charset = "gb2312"
*set viewer font name="Arial Unicode MS"

*****
* set options defined during installation:
*****

* To see other set: issue "set" command or look in help file (F1)
*****
set display variables=ON
set display databrowser=OFF
set output open=ON
* set options defined during installation:
*****
* Default folder defined as:
cd c:\epidata_report\TestData
set output folder="c:\epidata_report\TestData"
set language=english
set echo=ON

```

If we strip it of all comments, what remains is:

```

set echo=off
set viewer font size =12
set window font size =12
set editor font size =12
set viewer font name ="Verdana,Courier"
set display variables=OFF
set display databrowser=OFF
set output open=ON
set language=english
set echo=ON

```

a series of SET commands that tells EpiData Analysis some defaults that are operative until we change. We asked you before to write over this file and that replacing file differs from the above (comments that remain are not shown) is shown in red font:

```

1 set echo=off
2 cd temp
3 set display mainmenu=off
4 set display command prompt=off
5 set display worktoolbar=off
6 set viewer font size =12
7 set window font size =12
8 set editor font size =12
9 set viewer font name ="Verdana,Courier"
10 set style sheet="..\docs\en\epiout_monospace.css"
11 set display variables=ON
12 set display databrowser=OFF
13 set output open=ON

```

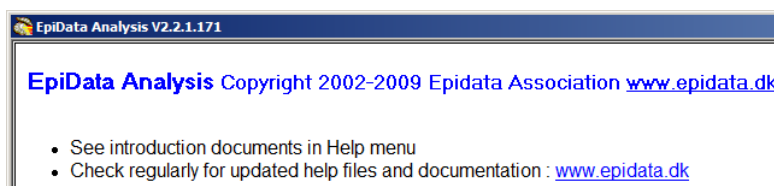
```

14 set output folder="\EpiData_report\temp"
15 set language=english
16 set echo=on
17 set start page="..\docs\en\start_tb.htm"

```

In Line 1, we direct EpiData Analysis to go to the (above created) sub-folder \TEMP. EpiData Analysis will be in this place when you start with an analysis and if you need to access data files or a program file or any other file, you will need to tell it where these are located *relative to this location*.

In Lines 2 to 4, we set the main menu, the process bar, and the command line off display. This is done for the purpose of this project because the end user is not to analyze the data, and only needs a plain interface. As a result, of these settings, the interface looks now as follows if you execute the program:



At the bottom, you see don't see anymore the command line, and only the sub-folder in which EpiData Analysis is after executing the first four lines:



The default style sheet that EpiData Analysis uses is the `output.css` file, a cascading style sheet. You could make the style in the `start.htm` file, the name of which we modify on line:

```

17 set start page="..\docs\en\start_tb.htm"

```

but the recommendation to the World Wide Web Consortium (W3C) is to refer in the main file to another specific (`*.CSS`) file that defines the style. This recommendation is for good reasons: you can always change the style sheet without changing the main HTML file. You copied the `epiout_monospace.css` file into the `\DOCS\EN` and Line 10 we instruct EpiData Analysis to use the latter style sheet instead of its default. Below are the results with the two respective style sheets. Do you note the difference?

Display with the default `output.css` file

Display with the
`epiout_monospace.css` file

Study country	
	N
Moldova	17725
Mongolia	22555
Uganda	54050
Zimbabwe	34478
Total	128808

Study country	
	N
Moldova	17725
Mongolia	22555
Uganda	54050
Zimbabwe	34478
Total	128808

Of course, you can make your own style sheets but this will require learning a bit more on how to do them, and is not subject of this exercise.

EpiData Analysis writes log files and other stuff that are useful to review when something goes wrong. These files are not usually used when all goes as expected and to get them out of the way, the command in Line 14 redirects the output to be stored in the sub-folder TEMP.

With this brief and rather simplified introduction of how EpiData Analysis starts to work, you should now have an understanding on what the `EpiDataStat.ini` file does and how you can always access it and change it on the fly. In your “regular” EpiData Analysis program, it will often prove useful to direct it with this file to a specific project folder on which you are currently working with the CD command in Line 1 and it will always be in the right place for the project until you change it.

After we are now done with the role of the `EpiDataStat.ini` file, we have to deal with the HTML file `start_tb.htm`, which gets us into learning some basics about the HTML language.

Making the interface in HTML

Using an HTML editor to make the skeleton of the interface

HTML is the language of the Internet which is interpreted by a browser such as the proprietary Internet Explorer™ or the open-source and free Firefox® to make it visually comprehensible and appealing for the user. A simple text that looks like:

This is a simple text to show the browser interpretation of HTML text and the actual underlying HTML.



We add above an icon for display.

has the following underlying HTML code in the browser has:

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2
3 <html>
4
5 <head>
6 <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type"><title>temp</title>
7 </head>
8
9 <body>
10 <big><span style="font-family: Arial;">This is a simple text to show the browser</span>
11 <br style="font-family: Arial;"><span style="font-family: Arial;">interpretation of HTML text and the actual</span>
12 <br style="font-family: Arial;"><span style="font-family: Arial;">underlying HTML.</span></big><br>
13 <br>
14 <big><span style="font-family: Arial;">We add above an icon for display.</span></big>
15 </body>
16
17 </html>

```

This is complex at first look, but when we look at it carefully then we realize that it is a highly logical language and sequence of instructions to the browser.

In Line 1, information is provided that the language conforms with W3C standards and the version of HTML it is using and that you can find this confirmed at the W3C website.

Every component in an HTML document has the principle to define the beginning and the end of the component and the system is the same for all. In the above example we have:

Begin	End
<html>	</html>
<head>	</head>
<body>	</body>

The structure indicates that everything between the opening and ending HTML tags is in fact HTML. Embedded are two parts, the head and the body, each indicating where it starts and where it ends. Every HTML page is build around these key parts, and within these you may have other sub-components imbedded that follow the same principles such as here within the body:

Begin	End
	

For the time being, we will leave it at that but will come back later to these principles when we work specifically on the `start_tb.htm` file.

Because the complexity for the beginner, a multitude of software has been developed that allows the user to write normally as in a word processor to see what one actually wants to get, and the software translates it into HTML and the page becomes interpretable by the browser. The advantages of such software are obvious but the downside is that such software is often very expensive (hundreds of Euros perhaps), and not all is adhering strictly to W3C commands which will make it difficult for some browsers that require strict adherence to interpret the language properly.

You may have heard about the Mozilla Foundation which produces free and excellent open-source software, such as the browser Firefox, the email client Thunderbird, the calendar Sunbird and yes, the HTML editor Nvu. Nvu is still in its infancy and has some problems, some of which have been resolved with the HTML editor KompoZer which you find on this course web in the software section and that we will be using.

You do not need to install KompoZer, it is just a zip file. Unzip it into the root of your hard drive and you get a folder:

KompoZer 0.7.10\

Make a shortcut to its executable file:

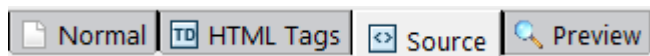
KompoZer.exe

to your desktop and if desired to the quick launch bar to have it accessible at your fingertip with one click away.

Access KompoZer and find the `start_template.htm` file in the `\DOCS\EN` sub-folder of the project. It is an empty page with a title:



That it is not quite empty becomes clear if you tick at the bottom to “Source”:



In fact, we have taken the normal `start.htm` file that comes with the installation of EpiData Analysis, have stripped it down and added a few essentials to prepare it for an interactive menu (template courtesy, Jens M Lauritsen, April 2008) and saved it under this `start_template.htm` file name.

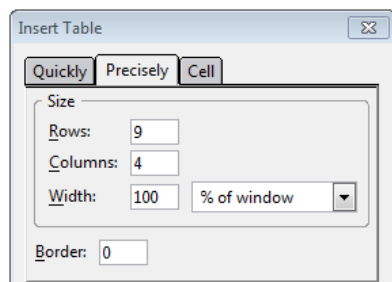
The first thing is to save it as:

`start_tb.htm`

Remember that we direct the `EpiDataStat.ini` file to go to this file when initiating:

```
set start page="..\docs\en\start_tb.htm"
```

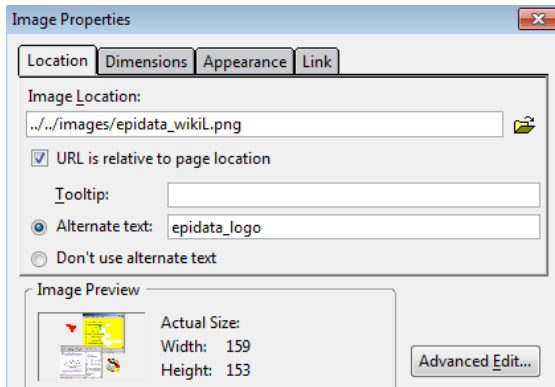
You will be working in the “Normal” tab and the first thing we do is to insert a table with 4 columns and 9 rows (you can count them in the screenshot of the final interface shown at the beginning). Choose “Precisely”:



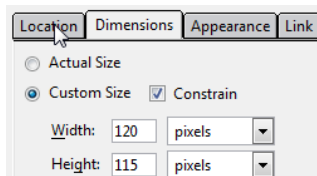
and set the Border to “0”. The default for “Border” is 1: a line around each cell is displayed in the browser. Setting it to “0” allows entering the information into cells instead of using the Tab key (which we cannot do properly in an HTML page), but the user does not see any line and remains unaware that we used a table.

If the table cell background is not white, you may have to fiddle with the options for them to render them white (non-transparent).

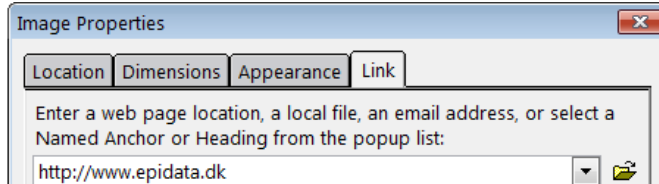
In the most upper left cell we insert one of the provided EpiData logos (the epidata_wikiL.png file from the \IMAGES sub-folder) using the “Insert” menu and also supply a (required) alternate text for the name:



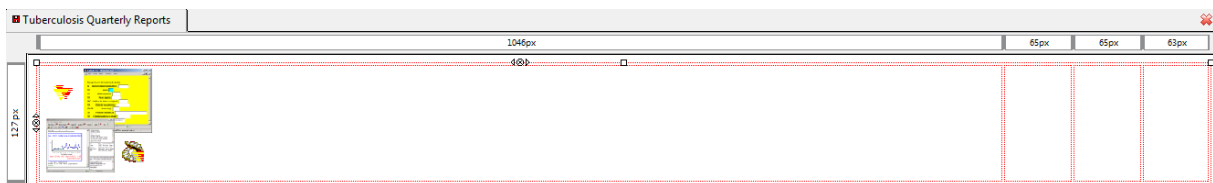
Then go to the tab “Dimensions” and decrease the current size of the width to 120 pixels:



In the “Link” tab add the URL of the EpiData website:



Then accept and you get:










Never mind for now the distortion of the column width. Add the Union logo into the most upper right cell and make a link to the Union website (<http://www.theunion.org>) in the same manner.

Join the central two cells of the first row, add the text and format it (see beginning of this Exercise) to get:



Continue adding text, formatting it, and add the appropriate icons into the appropriate places until you have the complete lay-out:

	<h1>Quarterly Reports on Case Finding and Treatment Results</h1>	
	Enter Data	
	Access EpiData Entry	
	Create Reports	
	A quarterly report on case finding	
	A quarterly report on treatment results	
	Terminate session	
	Erase temporary session files	
	Exit the menu	

Make sure to save the file. Perhaps you want to look at the source code and see that a lot has been added. We will not further edit the source code here, we will do this in a text editor. Thus exit KompoZer, this is all we are going to do with it.

If you now click the EpiDataStat.exe file, you will get:



Of course, there is no functionality yet (except the EpiData and Union web site icons if you are on the internet). In the next step we add functionality to the other icons.

Editing the HTML file

Windows has an inbuilt text editor, NotePad™. There are several free text editors available that prove to be more flexible and powerful than this one. Among the many, we have selected Crimson Editor™ as our choice for this course. This is a free and excellent text editor that

has several additional features such as showing HTML code in colors and providing the very useful feature for rectangular selections.

Open now in this text editor the `start_tb.htm` file. What you see may seem a bit overwhelming and we will thus take it apart into manageable pieces to understand what we find there and to edit it where appropriate.

You have seen before the first few lines:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html><head>
3 <meta content="text/html; charset=ISO-8859-1" http-equiv="Content-Type">
4 <meta name="copyright" content="EpiData Association">
5 <meta name="description" content="Introduction to EpiData reports"><title>Tuberculosis Quarterly Reports</title>
```

They inform us that we deal with an HTML file and that the head starts. If we collapse (and hide from view) lines 9 to 28, we see up to line 28:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html><head>
3 <meta content="text/html; charset=ISO-8859-1" http-equiv="Content-Type">
4 <meta name="copyright" content="EpiData Association">
5 <meta name="description" content="Introduction to EpiData reports"><title>Tuberculosis Quarterly Reports</title>
6
7 <style>
8 <!--
28 </style><!--<link href="epiout.css" rel="stylesheet" type="text/css" media="screen">--></head>
```

You note here the beginning and end tags for HEAD and those for STYLE embedded inside these:

```
<head><style>...</style></head>
```

Make two hard carriage returns after the closing `</head>` tag, so we see a bit better what we have next. Make another hard carriage return after the end of:

```
<body class="bodywhite">
```

so that we separate for visibility the beginning of the table definition:

```
33 <table style="text-align: left; width: 100%; border="0" cellpadding=.....
```

Can you see where our first row begins and where it ends?

```
<tr><td style="background-color: white;"><a href="http://www.epidata.dk"></a></td><td colspan="2" rowspan="1" style="background-color: white; text-align: center; width: 957px;"><big><big><big><big>Quarterly Reports on Case Finding and Treatment Results</big></big></big></big></td><td style="background-color: white;"><a href="http://www.theunion.org"></a></td></tr>
```

The row begins with the opening tag `<tr>` and its ending tag `</tr>`. Each cell embedded in the row has its beginning and end tags `<td ...></td>`:

	Begin tag	End tag
Row	<code><tr></code>	<code></tr></code>
Cell	<code><td></code>	<code></td></code>

If we isolate the first cell with the logo of EpiData, we have thus:

```
<tr><td style="background-color: white;"><a href="http://www/epidata.dk"></a></td>
```

<a href ...

is thus the opening about everything dealing with this logo: first the link to the web site, then information about the cell style, the dimensions of the logo, the alternate text, and finally the reference to the name of the image and the place where it is relative to the current position:

```
src="../../../images/epidata_wikiL.png"
```

Thus, in summary:

The definition of the image is defined as:

```
src=
```

and the image itself is given with its name relative to the location it had when you inserted it:

```
"../../../images/epidata.png"
```

We have no links to any of the programs that should be run when the user clicks on the EpiData Entry image defined as:

```
src="../../../images/epidata.png"
```

because there is not yet any instruction on what to do here. This is what we are going to do in the next step.

Opening EpiData Entry from within EpiData Analysis

The task to open EpiData Entry from within an EpiData Analysis program that runs when the icon on the report menu is clicked is approached in three sequential steps:

- 1) Writing the HTML code to invoke the EpiData Analysis program when clicking on the icon
- 2) Writing a batch file in DOS that opens a specific EpiData REC file
- 3) Writing an EpiData Analysis program that calls the batch file

1) Writing the HTML code to invoke the EpiData Analysis program when clicking on the icon

In the cell with the EpiData Entry logo we have the HTML code:

```
</td>
```

We will make an EpiData Analysis program that we call "run_entry.pgm". It will open EpiData Entry. We must now give the instruction in the start_tb.htm file to run this program if the user clicks the EpiData Entry logo.

Immediately before the `<img style` referring to the style of the `epidata.png` image (the EpiData Entry logo) we must introduce the reference to EpiData Analysis commands which are as follows:

```
<a href=' "epi:CLS;set echo=off;run " ../pgm/run_entry.pgm"; type "F8: Go back to menu"'><img ... .. </a>
```

You know already the `<a href`. The following EpiData Analysis commands are enclosed in single quotes ('). The `run` is the command to run the program which is in double quotes and defined by its relative location. Also note that forward slashes are always required for paths in HTML. What is missing is the closing tag for the `<a`. This comes write after the image definition, so that the complete code is:

```
<a href=' "epi:CLS;set echo=off;run " ../pgm/run_entry.pgm"; type "F8: Go back to menu"'></a>
```

The:

```
type "F8: Go back to menu"
```

is an instruction to return to the menu after the running of the instructions to open EpiData Entry (and leaving EpiData Entry) has been completed.

Summarizing some of the specific EpiData Analysis commands that we may use in HTML:

Action	EpiData Analysis command
Clear the screen and memory	<code>epi:CLS</code>
Show only output, not the running of the program	<code>set echo=off</code>
Do not display the main menu	<code>set display mainmenu=off</code>
Do not show the process bar	<code>set display worktoolbar=off</code>
Do not show the command line	<code>set display command prompt=off</code>
Run the program	<code>run " ../pgm/run_entry.pgm"</code>
Tell the user how to return to the main menu after the program has completed	<code>type "F8: go back to the main menu"</code>

Each of the commands must be separated by a semicolon.

This summarizes the principles we use to connect (making a link to) the execution of an EpiData Analysis program to an icon. While the HTML code is now correct, nothing will happen yet because the program `run_entry.pgm` does not yet exist. That EpiData Analysis program will have the instructions to execute a DOS batch file and we will make that batch file first, then the `run_entry.pgm`.

2) Writing a batch file in DOS that opens a specific EpiData REC file

A batch file is a simple set of instruction in DOS that is executed if the batch file is invoked. Batch files have the extension *.BAT. DOS is the operating that reigned PCs before the arrival of the Windows operating system and is still available under the hood of Windows.

Our batch file has a very simple structure. It does nothing else than telling where the file that is sought is located and that it has to be started. In the EpiDataStat.ini file we instructed EpiData Analysis to start at opening in the \epidata_report\temp folder. You placed the empty (no data):

```
sample_2006.chk  
sample_2006.rec
```

files into the \epidata\originals folder. Therefore, the batch file is simply:

```
cd originals  
start sample_2006.rec
```

It directs to change the folder and then open the program that recognizes a file with the extension *.REC. Because we have set the EpiData Entry defaults to associated *.REC files with EpiData Entry, this is the program that will open.

Save this file as “startepidataentry.bat” in the root of the \epidata folder.

3) Writing an EpiData Analysis program that calls the batch file

Because we have set up EpiData Analysis in the \epidata_report folder in a way that shows no tool bars, no command line, and the echo is set off, etc, we cannot reasonably develop out programs with the “Report EpiData Analysis” program. Instead, we will use our “Normal EpiData Analysis” program. To this end, we should simulate the folder structure of the \epidata_report folder. Instead of the \epidata_report folder we use the c:\temp folder. Empty it of any content it may currently have and make the necessary subfolders that we need, i.e.:

```
\data  
\originals  
\pgm  
\temp
```

and copy the contents of the from the \epidata_report\originals folder to the \temp\originals. This set-up will assure that we can properly testing the functionality of our relative paths. In the epidatstat.ini file of our “Normal EpiData Analysis” we must thus also set our starting folder analogously to \temp\temp.

Once you have assured functionality in our “Normal EpiData Analysis”, we copy the *.PGM file over to the \epidata_report\pgm folder, close the “Normal EpiData Analysis” program and open the “Report EpiData Analysis” program.

We will call our program “run_entry.pgm” and save it in the \temp\pgm sub-folder. Again, the present position is a sub-folder of \temp and we always want it to be in a sub-folder at the beginning, so that the command:

```
cd
```

gets us one level up to the `\temp` or later to the `\epidata_report` but never out of that “container” (thus the relative path will never be violated and get into conflict with the drive or folder).

It might be a good idea to interactively prompt the user to confirm that he or she really wants to switch to EpiData Entry, so we build such a prompting into it:

```
cls
close

cd ..
cd originals

define yesno # global

yesno=?Confirm to start data entry: 1=Confirm - 2=Stay in report?
  imif yesno=1 then
    cd ..
    ! "startepidataentry.bat"
    cd originals
  else
    cls
  endif
```

Let's take this apart.

```
cd ..
cd originals
```

get's us from `\epidata_report\temp` one level up to `\epidata_report` (note the actual `\epidata_report` folder is used here interchangeably with the simulation `\temp` folder) in the first line, and then one level down to `\epidata\originals` in the second line. This is not really necessary here, the main thing is that we are certain to be in a sub-folder of `\epidata_report`, and this is the one where we keep the data files.

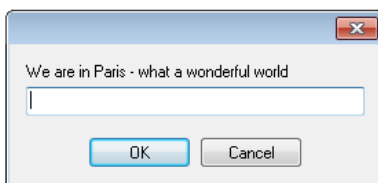
```
define yesno # global
```

This creates a numeric variable in the memory. The command `global` will assure that it will be retained following a `close` command. Such variables are like constants. It is not really needed here, but it is handy to know as you will see later.

Interactive prompting interrupts a program and requires the user to do something before it continues. Typing:

```
?We are in Paris - what a wonderful world?
```

gives you a pop-up box:



Thus, what is shown on the output screen is placed between two question marks:

```
yesno=?Confirm to start data entry: 1=Confirm - 2=Stay in report?
```

The user is requested here to assign a value of 1 or a value of 2 to the just created variable `yesno`.

The command `imif` is used to divert the program, according to the value that was just assigned before, thus:

```
yesno=?Confirm to start data entry: 1=Confirm - 2=Stay in report?
imif yesno=1 then
  cd ..
  ! "startepidataentry.bat"
  cd originals
else
  cls
endif
```

Thus, if the user entered a value 1 for the variable `yesno`, it will move up one level to `\epidata_report` with the command:

```
cd ..
```

The command:

```
!
```

opens a DOS window, executes any valid DOS command and returns to EpiData Analysis. *Note that this works only for Windows XP and higher.* The `*.bat` in the `startepidataentry.bat` identifies the file as a DOS command to execute this particular batch file, located in the `\epidata_report`. (You must of course copy it over to the “Normal EpiData Analysis” program to test functionality).

If the value for `yesno` was not 1, i.e. 2, then the screen is cleared, EpiData Analysis stays where it was, and the program completes.

In summary what will happen with the above:

- 1) The person clicks the EpiData Entry icon, this invokes the `run_entry.pgm`
- 2) In the `run_entry.pgm`, the user is interactively prompted to enter a value that determines whether a batch file is called or not. If the batch file is called by giving the value 1 to the variable `yesno`,
- 3) the `startepidataentry.bat` file is executed in DOS.
- 4) As we have set the EpiData Entry “`epidata.ini`” to associate any `*.REC` file with EpiData Entry, EpiData Entry opens the file “`sample_2006.rec`” and the person can enter data only into this file. We have slightly edited the “`epidata.ini`” file, so that it does not show the process bar to reduce the risk that the entry person starts fiddling with it.

- 5) Once data entry is completed, the instruction on the top on the entry questionnaire tells the user to close the file with F10, and then to quit the program again with F10.
- 6) Once EpiData has been quit, the interface is again that of EpiData Analysis, with the information to press F8 (see above) to return to the main interface and the circle is closed.

Checking functionality in the “Report EpiData Analysis” and add links to the other programs in the `start_tb.htm` file

After you have completed the `run_entry.pgm` and checked its functionality in the “Normal EpiData Analysis” close the latter and copy the program file to the `\epidata_report\pgm` folder, then run it interactively from the menu in the “Report EpiData Analysis”.

You will make a total of four programs:

```
run_entry.pgm
sample_1_quarterly_report.pgm
sample_2_quarterly_report.pgm
cleanup.pgm
```

It might be best to edit the `start_tb.htm` file now for the remaining icons to link them to the three remaining programs, so that you are done with the HTML file. The principle is exactly the same as shown above for the `run_entry.pgm`, shown below for the “`sample_1_quarterly_report.pgm`”:

Insert the necessary code into the “`start_tb.html`” file

In analogy to the above with the program that starts the choice to switch to EpiData Entry, we look for the code where we have the reference to the first EpiData Analysis program:

```
</td><td style="background-color: white;
width: 957px;">A quarterly report on case finding</td>
```

and as above we must insert the reference to the program immediately before the `<img` tag and not to forget the ending `` tag:

```
<a href="epi:CLS;set echo=off;run "../pgm/sample_1_quarterly_report.pgm";
type "F8: Go back to menu"></a></td><td style="background-color:
white; width: 957px;">A quarterly report on case finding</td>
```

Making the EpiData Analysis program to produce the quarterly report on case finding

We have two actual datasets of a random selection of tuberculosis case registers from the years 2004 and 2005 from Viet Nam. They were provided by courtesy of Dr Nguyen Binh Hoa from the National Tuberculosis Program Viet Nam. They were slightly edited from the original case registers in that the original identifier was removed and replaced by an artificial one. The names of the 30 units were also changed to give a simple non-reality coding.

Several variables that were collected were also removed and some other minor modifications were made. However, overall, these data are real.

The task is to produce a quarterly report on case finding. We follow here the guidelines of The Union for the quarterly report. It requires that all notifiable cases in the quarter are reported. We will get back to this shortly.

It may be desirable not only to get an output for a given quarter, but for an entire year. A simple way of offering this option is to prepare very small EpiData dataset with just three records and two variables (a minimum of two variables is required by EpiData). The two variables are:

```
number Sequential number of year #
choiceyr      Year of selection #####
```

The years for selection (three records) are 2004, 2005, 2006. Note, however, that 2006 has no data. For your convenience, the files:

```
choice_year.qes
choice_year.rec
```

have been included among the required files. In analogy to what was shown before, one would start the program (save it as "sample_1_quarterly_report.pgm" in the \epidata_report\pgm folder after testing it in the corresponding simulation folder) with:

```
cls
close

cd ..
cd originals

read "choice_year.rec"
cls
define yr ##### global
cls
set echo=on
freq choiceyr /v1
type "Note: you can select 2006 only if you have entered data" /h2
yr=?Define year for analysis?
set echo=off
```

This determines the year of choice. Nevertheless, one still preferable appends all three years in a single dataset and then makes the selection based on the above choice. Thus, the program would continue with:

```
cls
type "Be patient...files are identified and prepared" /h2
close
read "sample_2004.rec"
append /file="sample_2005.rec"
append /file="sample_2006.rec"
select if regyy=@yr
savedata "temp01.rec" /replace

type "Be patient...files are identified and prepared" /h2
```

Because we are setting `set echo=off`, the above line is written to appear on the screen to inform the user that something is running even if nothing is seen.

Note here the line:

```
select if regyy=@yr
```

The `@yr` refers to the global variable defined above and as mentioned earlier it is still present (like a constant) despite the closing of the “`choice_year.rec`”.

While one has the registration month, it is the quarter that is of interest, thus this variable must be defined:

```
cls
close

read "temp01.rec"

define quarter #
if regmm>=01 and regmm<=04 then quarter=1
if regmm>=04 and regmm<=06 then quarter=2
if regmm>=07 and regmm<=09 then quarter=3
if regmm>=10 and regmm<=12 then quarter=4
label quarter "Calendar quarter"
labelvalue quarter /1="Jan-Mar"
labelvalue quarter /2="Apr-Jun"
labelvalue quarter /3="Jul-Aug"
labelvalue quarter /4="Sep-Dec"
```

We need more variables. The following two parts are required for the quarterly report as recommended by The Union:

ALL CASES REGISTERED IN THE QUARTER

SMEAR-POSITIVE				SMEAR-NEGATIVE		EXTRA-PULMONARY	TOTAL
New cases	Relapses	Treatment after failure	Treatment after default	< 15 yrs	15 + yrs		

NEW SMEAR-POSITIVE CASES ONLY

Age group (years)														TOTAL		
0-14		15-24		25-34		35-44		45-54		55-64		65 +		Male	Female	Total
M	F	M	F	M	F	M	F	M	F	M	F	M	F			

A note to the notifiable cases: Patient categories “Transfer in” and “Other” must not be reported.

When you define your variables, note that not all registered cases may have always the required information. For instance, you need three variables to determine “SMEAR-NEGATIVE, <15 yrs”: The case must be pulmonary, you must know the case is smear-negative, and you must know the age. Take this into account when deciding what you are going to present the analysis.

If you want to produce a graphics output of two graphs, and show them side by side. You must first save the graphs, e.g.:

```
bar repdef2 \  
    /ti="All cases" \  
    /save="all_cases_1.png" /replace  
cls  
  
select repdef2<>0  
select repdef2<>9  
bar repdef2 \  
    /save="all_cases_2.png" /replace  
cls
```

You have received among the required files and HTML file “side_by_side_graphs.html”:

```
echo <table><tr><td colspan=2 border align=center >  
<font color=black face="Arial" size="4">Some title that crosses over both  
graphs</font></td>  
<tr><td></td><td></td></table>
```

Replace the example names with the actual names of your graph files and then the simple command in EpiData Analysis:

```
show "side_by_side_graphs.html"
```

will show them side-by-side.

Other programs

The process is analogous for the quarterly report on treatment results. You can use the same basic program up to the point where you produce the actual output. You may consider two outputs, one for all cases, and one for sputum smear-positive cases only.

It might be useful to have a separate program to erase all files created during the session, so that only the files remain that are needed to start anew.

Commands to exit the program

To exit the program, you do not need a special program. The EpiData analysis command for the HTML file to quit the program without asking for conformation is:

```
<a href='"epi:exit'> ... </a>
```

Task

Produce a single folder containing the EpiData Analysis program, including the database of with an interface that permits by simple clicking to enter data or to run two standard reports, giving the user the option of choosing which country, year, laboratory, etc to

analyze and is transferable to any drive or folder, with a total zipped file size of just 3.81 MB.

A user will be able to unzip this file onto any drive or folder, double the \epidata-report folder, look for the epidatastat.exe file, double-click it, and being in it. Test it out by zipping it and then unzipping it onto a flash USB stick.