User manual

Version 2.0

(October 2011)

Post-it® is a 3M product.
Adobe Reader® is an Adobe Systems Inc. product.
Delphi 2007® is a CodeGear/Borland Software Corporation product.
MsDos® and Windows® are Microsoft Corporation products.
Eyelink®, Eyelink®II and Eyelink®1000 are S.R. Research Ltd. products.
ASL504 ® is an Applied Science Laboratories product.
EyePuter ® is an Electronica, S.A. product.
iViewX is a SensoMotoric Instruments GmbH (SMI) product.
Intuos, Intuos2, Intuos3, Intuos4 & Cintiq® are a Wacom Company Ltd. Product.

This software uses Indy components, Copyright (c) 1993 - 2003, Chad Z. Hower (Kudzu) and the Indy Pit Crew - http://www.IndyProject.org .

This software uses icons drawn from the special glyFX Borland Image Library Pack - http://www.glyfx.com.

This manual is not a manual for the above-mentioned products.
Its content does not involve their respective manufacturers in any way.

Table of contents

Figure list

CHAPTER 1: Installation and configuration

INTRODUCTION

Writing is a complex human activity. The writer has to compose a coherent message and formulate it in accordance with linguistic rules (grammar, spelling), all the while taking the characteristics of the potential reader into account. The ability to manage all these various mental activities, as well as their time course, can be regarded as an indicator of the writer's expertise (Alamargot & Chanquoy, 2001). For cognitive science researchers, identifying the rules that govern the engagement and course of these mental processes is an essential step towards a greater understanding of writing and processing (Levy & Ransdell, 1996). The Eye and Pen software was designed to help researchers attain this goal (Chesnet & Alamargot, 2005; Alamargot, Chesnet, Dansac & Ros, 2007).

Eye and Pen can be mainly (but not exclusively) used in the context of handwriting studies, whatever the graphic format (from text to drawing).

The 'Eye and Pen' software was designed to allow the synchronous recording of handwriting (by means of a digitizing tablet: coordinates and state of the pen) and eye movements (via an optical eye-tracking system: eye coordinates in the task environment). The conjunction of these two signals allows us to study the synchronization between eye and pen movements during pausing and writing periods. For instance, it makes it possible to study not only the visual control of graphomotor execution, but also the reading of the text in order to revise it and the consultation of documentary sources with a view to summarizing them. Eye and Pen allows users to conduct these investigations in a continuous way, without interrupting the activity underway or increasing cognitive load.

In short, this software makes it possible to establish a link between the visual input of writing (gazes on documentary sources and/or text produced so far) and its graphomotor output (pausing and writing phases, drawing). This link can be studied in various situations and media (paper, computer screen, screen tablet, etc.)

The Eye and Pen software can be regarded as a type of digital video recorder, which allows users to watch and play-replay the process of graphic generation and associated eye movements as often as they wish. A semi-automated coding system enables users to characterize and classify ocular and graphomotor events.

Eye and Pen has two different modes of operation:

- the acquisition mode, allowing the recording of tablet and eye tracker data;
- the analysis mode, allowing the selection (with various filters) and editing (in text format) of data.

Obviously, using an eyetracking device is not mandatory, and you can choose solely to record and analyze written production.

Note: on our Website http://www.eyeandpen.net you will find all the latest information, updates, a forum and much, much more besides.

GENERAL REMARKS

In this section, we introduce a number of notions that are used throughout this manual. To make the latter less cumbersome, they are not defined elsewhere.

1. Real time experiment

Recording a huge amount of data in real time requires the computer managing this task to be 100% available. Evidence suggests that this is not the case of a computer running Microsoft Windows.
Nevertheless, it is possible to optimize recording conditions, by disabling antivirus software, automated updates and on-line chat applications (msn©, aim© etc.), unplugging or deactivating unnecessary devices, etc.

<u>Caution</u>:

Windows **themes** and **special effects** (blinks, sounds, etc.) may reduce acquisition quality by creating unforeseeable system response delays.

Switch to Windows "classic mode" (Windows 2000/XP) and/or turn off Windows themes (or Aero).

Windows Firewall will prevent Netsync from working properly. We advise you to disable this firewall for the duration of the data acquisitions.

The Eye and Pen Web site (`http://www.eyeandpen.net`) provides a number of hints on this topic (see FAQ section, Acquisition).

2. Picture files

Eye and Pen supports the BMP, EMF, GIF, ICO, JPG, JPEG and WMF file formats wherever a picture file may be required (configuration, acquisition, script, etc.).

<u>Note</u>
The BMP file format is the only one that is not compressed. This means that saving a given picture in BMP format will result in a greater file size than it would in other formats (e.g. JPG), but its loading time will be shorter because there will be no uncompressing. This fact should be taken into account when choosing file formats for an experiment.

When a file is selected via a dialog box, the latter will show a preview of the selected picture.

<u>Important</u>: the bigger the picture and the greater the number of colors, the longer it will take to load and display.
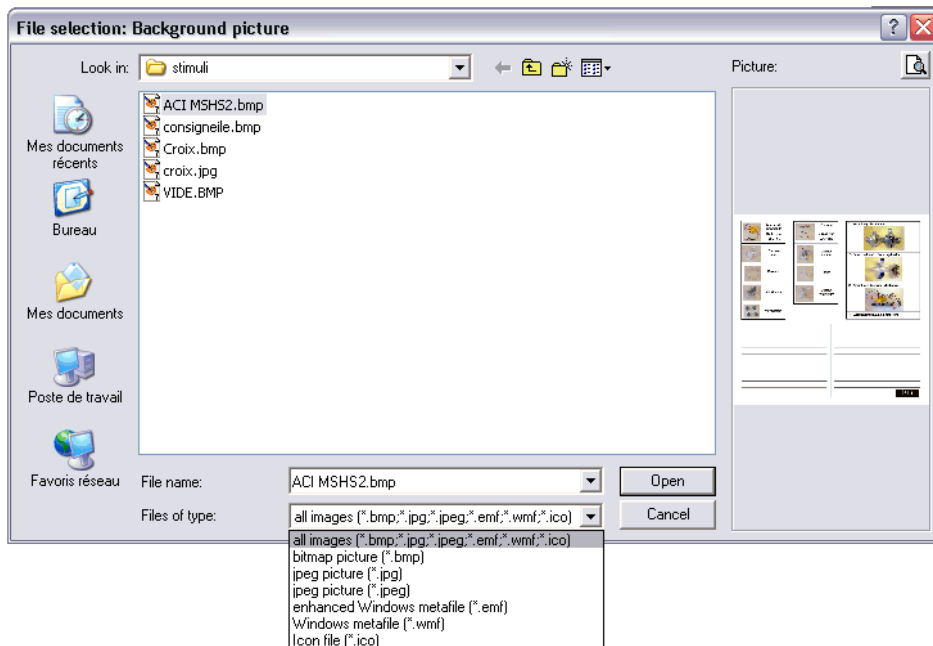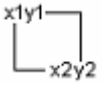
Figure 1: Picture selection dialog box (Windows XP).

3. Tablet coordinate representation

This representation is also applicable to "screen" coordinates and to the definition of rectangles.

LABEL	DESCRIPTION
	X1 and Y1 coordinates correspond to the **upper left-hand corner** of the tablet's active area. X2 and Y2 coordinates correspond to the **lower right-hand corner** of the tablet's active area.
X1	Coordinate of the left edge of the tablet's active area. This point is the horizontal **coordinates origin**.
Y1	Coordinate of the upper edge of the tablet's active area. This point is the vertical **coordinates origin**.
X2	Coordinate of the right edge of the tablet's active area (equal to the width of the tablet's active area).
Y2	Coordinate of the bottom edge of the tablet's active area (equal to the height of the tablet's active area).

To select coordinates on the tablet, click on the icon.
A dialog box shows up, with 4 fields containing coordinates values.
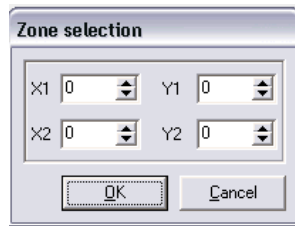
Figure 2: Selecting coordinates on the tablet.

Press the pen in the upper left-hand corner of the area you wish to select on the tablet (X1Y1). Keeping the pen pressed down, slide it to the lower right-hand corner of that area (X2Y2), then lift it up.

The zone (area) coordinates will be updated in the dialog box fields. You can restart this selection process as many times as you like.

When you are satisfied, click on the "OK" button to validate.

Hint: to define a zone on a tablet, you can also stick a "Post-it®" (or something similar) onto the tablet, or even draw the desired zones in a picture. Then, either use this picture as a background picture ("*background picture*" option of the "*display*" tab in the acquisition configuration panel if you are using an LCD tablet), or print it out and lay it on top of the tablet.

4. Relative file paths

In a number of script commands, you will have to mention filenames.

By default, the software will look for pictures in Eye and Pen's "stimuli" folder (or in the folder you selected in your configuration or redefined through a script command).

However, for reasons of your own, you may want to designate a file that is not in the default folder. There is a solution: the relative path.

An example:

```
SetUnmaskFile(..\cursors\Mask_oval_asym_right170.bmp)
```

Explanation

Folder structure:

Eyeandpen
    |-> Stimuli
    |-> Cursor

The picture file is located not in the Stimuli folder but in the Cursor folder. Accordingly, the relative path will start from the Stimuli folder:

1. " .. " allows you to move up to the next level in the folder tree, i.e. to leave the "Stimuli" folder and move to the "Eyeandpen" folder.
2. " \ " allows you to move down to the "Cursor" subfolder.
3. " \ " allows you to designate the picture file contained in the "Cursor" folder.

Relative paths from a default folder can be used anywhere that a file has to be loaded from a default folder.

Note: in this manual, the terms "folder" and "directory" mean exactly the same thing.

I. HARDWARE

I.1. Requirements

Component	Requirement
System	Windows 2000, Windows XP, Windows XP TabletPC edition, Windows Vista, Windows Seven. You will have to use a user account with administrative rights to be able to install and use Eye and Pen.
Processor	PII 900 MHz or better.
RAM	Min. 512 MB RAM (min. 1024 MB with Windows Vista).
Free disk space	50MB for a complete installation (with samples), plus space needed for recordings and analyses.
Video adapter	Min. 16 MB of video memory. Min. resolution 800x600. For greater comfort, we recommend 32MB and a resolution of 1024x768.
Tablet	A Wintab32-compatible driver (see <u>Appendix</u> for a list of manufacturers). This driver will normally have been supplied with your tablet.
Others	A free USB slot is required for the SafeNet Sentinel dongle (security lock key). Eye tracker (optional): "Eye & Pen" supports the following eye trackers: • Applied Science Laboratories ASL504 (serial) ; • Alphabio EyePuter ; • S. R. Research Eyelink, Eyelink II et Eyelink 1000; • S.M.I. iViewX

I.2.   Tablet installation

USB, serial or Bluetooth connection
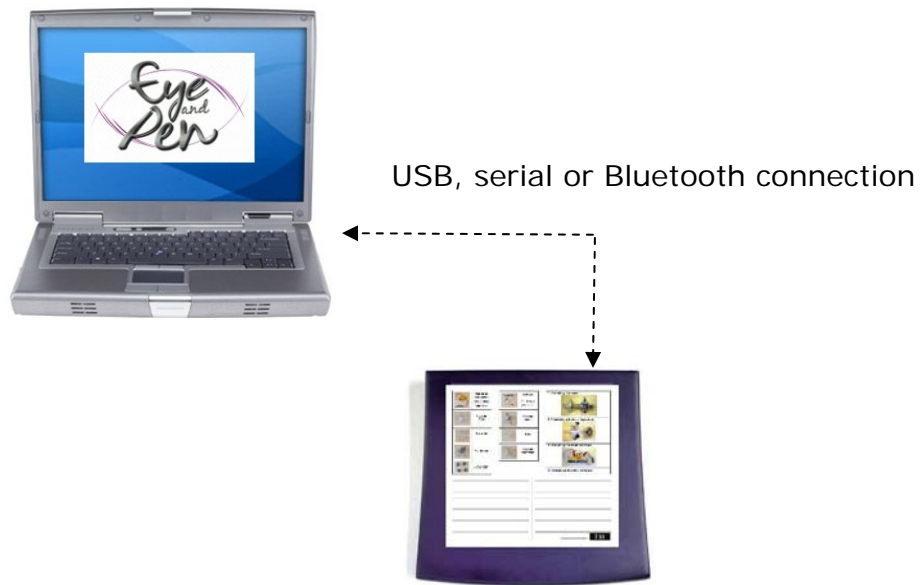
Figure 3: A tablet connected to a laptop.

STEP	DESCRIPTION
1	Switch your computer on.
2	Connect the tablet to your computer (follow its user manual instructions). **Important**: if it is to be taken into account in "Eye and Pen", the tablet must be **Wintab32-compatible.** (see appendix)
3	**Install your tablet driver**. Set your tablet up following the instructions in the **user manual** supplied with your tablet**.** **Simply plugging the tablet the fact that it can be used in Windows is not enough to have it to work properly in Eye and Pen. The tablet's driver is to be installed anyway.**
4	Set the tablet **in absolute mode.** The tablet will then return the pen's "true" coordinates, rather than a relative position. The "absolute mode" option may vary from one tablet to another. For Wacom tablets, for instance, the name of the option is "stylus mode". Check your tablet user manual. To obtain the maximum sampling rate, select the "**recognition data**" option (no dual track). Here again, the name of this option may vary, so check your tablet user manual.

I.3. Eye tracker installation

Following the instructions in the eye tracker's manual, connect it (or its data transmission device) to the computer executing Eye and Pen.
For example, Eyelink will be linked to your computer through a network cable, whereas ASL504 will transmit its data via a serial cable connected to a free COM port on your PC. Read the manufacturer's instructions carefully and use your eye tracker in accordance with its guidelines, including the safety instructions.

Note: to support Eyelink I, users must update the interface library (Eyelink Display Software) to version 2 (contact S.R. Research if in doubt).
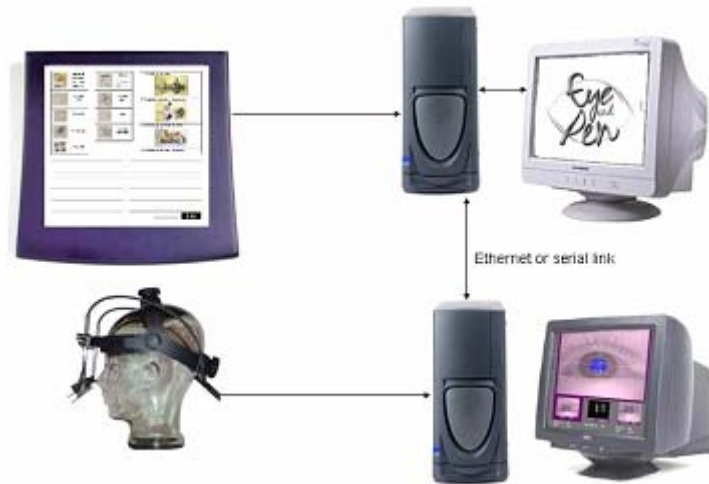
Figure 4: The Eye and Pen PC "drives" the tablet and receives the eye-tracking data.

I.4. Eye & Pen installation

STEP	DESCRIPTION
1	Insert the CD labeled "Eye and Pen" into your CD device. The Eye and Pen setup should automatically begin. If nothing happens, launch Setup.exe (on the CD-ROM root directory) and follow the instructions.
2	Insert the USB dongle (key) supplied with "Eye & Pen" into a free USB port.
3	Follow the program instructions to install Eye and Pen. It may also try to install the dongle driver. **Important**: if the dongle is not plugged in, Eye and Pen will run in "Demo mode" and you will not be able to record data, save analyses or edit results.
4	Launch the software: Start menu/Programs/Eye and Pen 2/EP. If you see a warning message about the dongle (SafeNet Sentinel) : • check that the dongle driver has been installed (you may need to re-install it) ; • check that the dongle has been plugged in. If the problem persists, leave a message describing the problem (including the error code) on the forum (see Error codes).

I.5.   Starting Eye and Pen

There are three ways of launching Eye and Pen:

- click on "Start", select "Programs", then "Eye and Pen 2" folder. Click on "Eye and Pen" to start the program;

- go into the "Eye and Pen" software directory (default is c:\Program Files\Eye and Pen 2) and click on "EP.exe".

- on subsequent uses of Eye and Pen : double-click on a data file in Windows Explorer (filename has a extension TAB) or on an analysis file (filename has a TWK extension).

When launching it for the first time, the configuration panel will be displayed to let you select the language (drop-down list in the frame labeled "*language*")

Figure 5: Selecting the application language.

Command line:

The program may also be launched through a command line (Start menu/Execute or in a "command prompt" window), with or without options.
Type in the following command:

*<hard drive name>***:***<software directory path>***\ep.exe**   *[option1] [option2]*

Options are:

[-*f <.tab ou .twk filename>]*
        Directly opens a "file.tab" file (for a new analysis) or a "file.twk" file (to resume an ongoing analysis).

[*nosplash*]
        Do not display Eye and Pen splash screen at start (launch is faster).

Example 1: to open the "participant1.tab" file without the Eye & Pen logo.
```
 C:\Program Files\Eye and pen 2 –f c:\mydata\participant1.tab
nosplash
```

Example 2: to re-open "participant23.twk", the analysis of participant23 (Eye & Pen logo will be displayed).
```
 C:\Program Files\Eye and Pen 2\ep.exe –f
c:\mydata\participant23.twk
```

The order of the options does not matter.

II.   DEVICE CONFIGURATION

(*File/configuration/acquisition* menu)

The acquisition configuration panel has five tabs, allowing users to configure:

- the tablet being used;

- the connected eye tracker;

- the "Simple" acquisition paradigm ;

- script-based data acquisition;

- script-based data acquisition, network synchronized;

- the display parameters.

To launch the acquisition configuration panel, click on the "*File*" menu, select "*Configuration*" and click on "*Acquisition*".
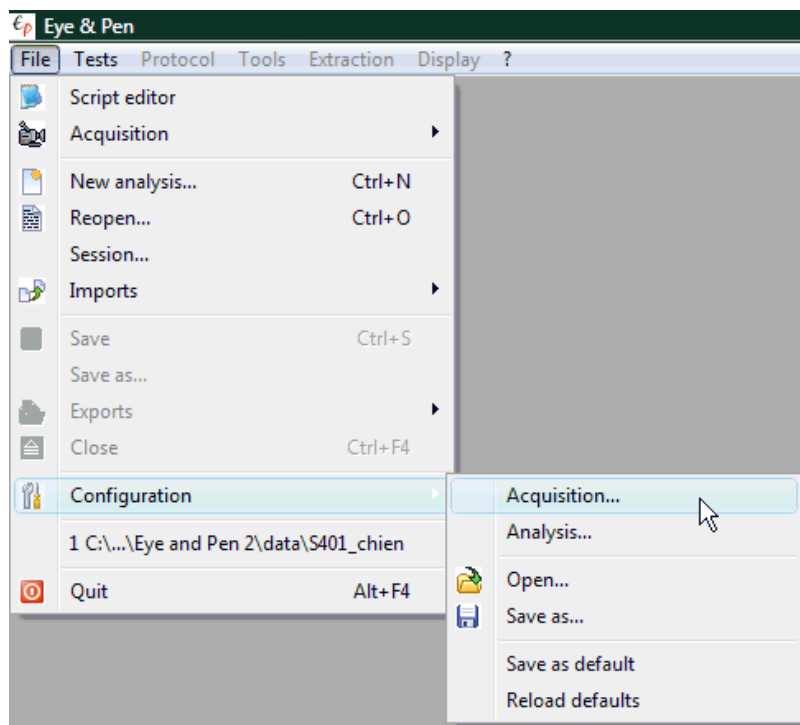
Figure 6: Launching the acquisition configuration panel.

Note: for the sake of data consistency, this panel cannot be launched when a protocol analysis is in progress.

II.1. Tablet configuration

(*File/configuration/acquisition* menu, "*Tablet*" tab)
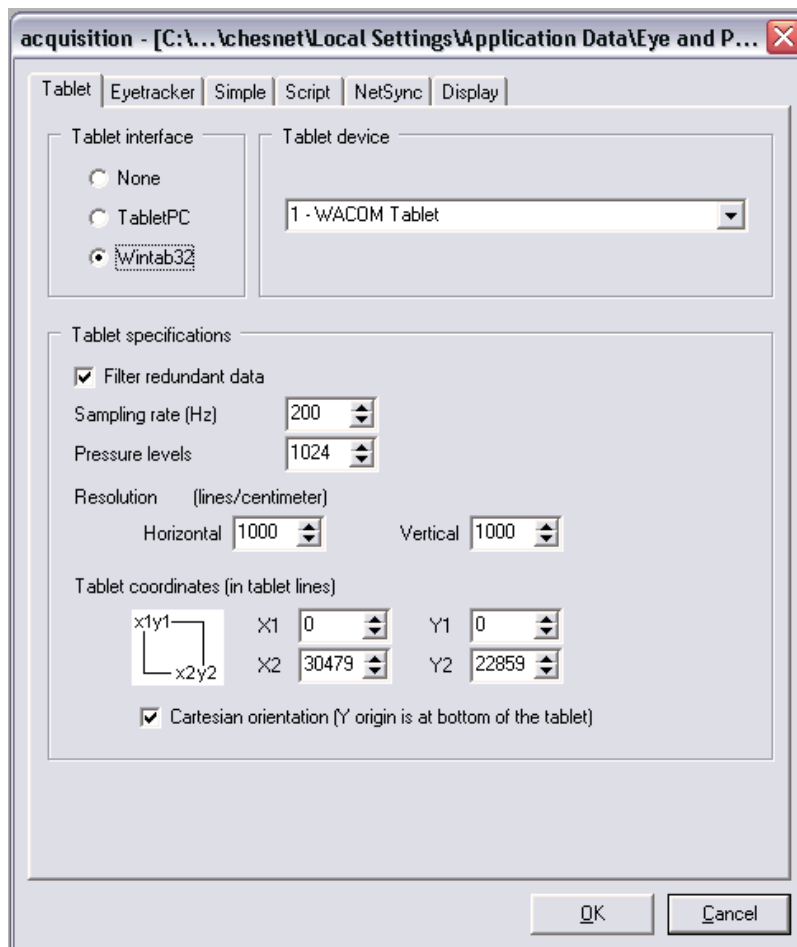
Figure 7: Acquisition configuration panel, "Tablet" tab.

LABEL	DESCRIPTION
Interface	Choose the type of tablet: • None: an acquisition can be performed without a tablet. • TabletPC: you will use the internal screen of a Tablet PC • Wintab32: generic interface for external tablets connected to the computer.
Tablet	Allows you to choose a tablet amongst those recognized by the Wintab32 driver. Connected tablets are numbered from zero upwards. **Important**: when a tablet is selected, its frequency, resolution and active area coordinate fields are automatically filled in. You can manually modify these values.
Filter redundant data	If this option is ticked, the program will not record identical tablet data (successive samples with the pen at the same place with the same pressure).

Sampling rate (HZ)	The tablet's sampling rate represents the number of times per second that pen data (coordinates and pressure) are sent by the tablet. **Note**: for Tablet PC, its value is always 133 Hz because the system does not always return a true value. You may use the performance test to empirically fix this value (p. 36).
Resolution	These fields show the selected tablet's horizontal and vertical resolutions. You can manually change the values. These values are involved when converting calculation results into measurement units (mm, cm, inches, etc.).
Pressure levels	Range of the pressure measurement scale.
Tablet Coordinates (in tablet lines)	This option shows the tablet's active area coordinates. You can manually change theses values. If you are using a **double** screen (LCD tablet plus a standard monitor), see Appendix.
Cartesian orientation (Y origin is at bottom of the tablet)	This option allows you to choose the type of coordinate system used by the tablet (check your tablet's user manual). If your tablet uses a **Cartesian system of coordinates** (the origin of coordinates is located in the lower left-hand corner of the tablet), check this box. The program will then update the tablet data to align it on the screen coordinates system, which has its origin in the upper left-hand corner. If there is an error, the tablet data will be displayed "upside-down". **Standard tablets** (non-LCD) generally use a **Cartesian system, whereas LCD (Liquid Crystal Display) tablets** use a **screen coordinates** system (as they are also a screen)**.** This option is checked by default for Wintab32 devices and unchecked for Tablet PC.

II.2. Eye tracker configuration

(*File/configuration/acquisition* menu, "*Eye tracker*" tab)

Once the acquisition configuration panel has appeared, click on the "*Eye tracker*" tab.
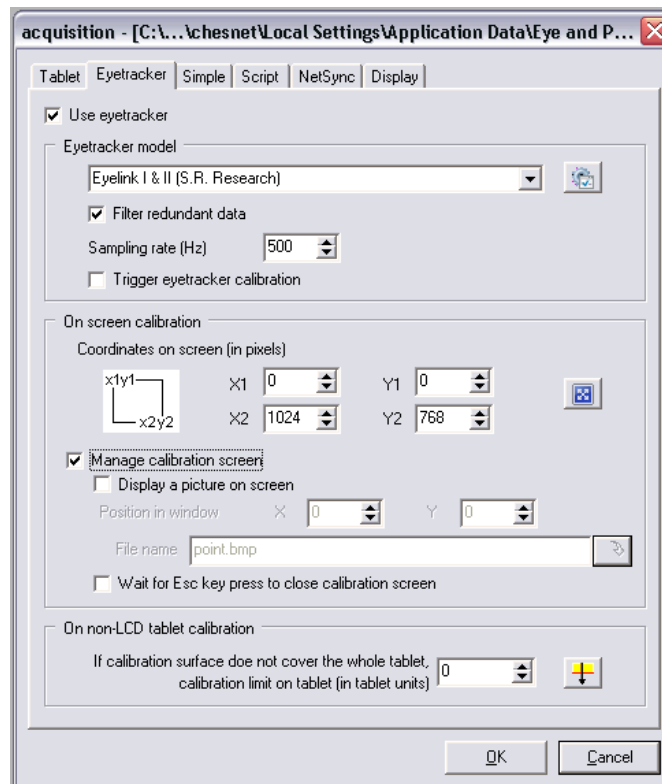
Figure 8: Acquisition configuration panel, "Eye tracker" tab.

The "*Eye tracker*" tab shows three frames:

- "*Eye tracker model*": choose the eye tracker you are using from the list;
- "*On screen calibration*": for calibration on a PC screen or an LCD tablet;
- "*On non-LCD tablet calibration*": for calibration on a standard tablet.

II.2.1. Eye tracker model

LABEL	DESCRIPTION
Use eye tracker	This option allows you to choose whether you want to use an eye tracker or not. If this option is selected (ticked), the "*Eye tracker model*" options become available. You can select the eye tracker model from the drop-down list below the option. Four models are available: **Eyelink I and II, ASL504, Eyeputer, iViewX** and **Tablet.** **Important:** if you wish to use an eye tracker that is not listed, leave a message on the Eye and Pen forum.

![icon]	This button allows you to set the eye tracker parameters. If you are using: • Eyelink I and II: see page 24; • EyePuter: see page 27; • ASL 504: see page 26; • iViewX: see page 29; • Tablet: see page 31.
Filter redundant data	This option allows you to discard successive items of identical eye data (the eye is still watching the same location). With a very accurate and high-speed sampling eye tracker, this option is not particularly useful, as our eyes move slightly all the time. On the other hand, a less accurate eye tracker with a low sampling rate may return many identical eye positions when the eye is in a "stationary" state.
Sampling rate	Represents the number of times per second that the eye tracker samples the eye position. This value has to be set manually, as many eye trackers do not return this value, and some may have more than one sampling rate available.
Trigger eye tracker calibration	Allows you to manually launch the eye tracker calibration. This option has no effect for EyePuter and ASL504, they do not have a remote "triggerable" calibration procedure.

II.2.2. Eye tracker configuration ("Parameters" button)

a)   Eyelink I and II (S.R. Research)

(*File/configuration/acquisition* menu, "*Eye tracker*" tab, "*Eyelink I and II*" model, then "*Parameters*" button)
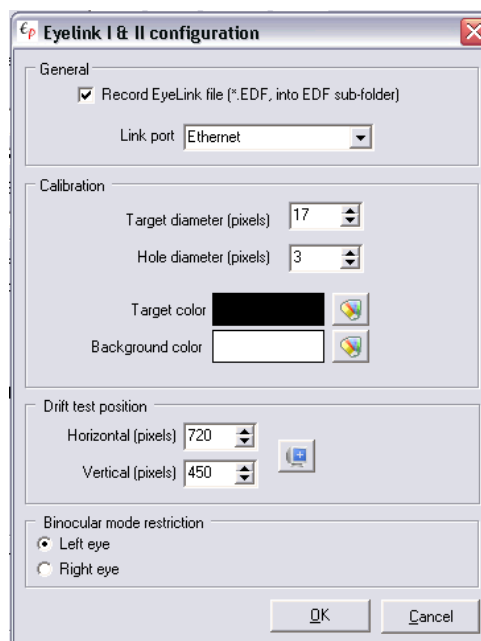
Figure 9: Eyelink parameter configuration.

The configuration panel for "Eyelink I and II" contains four frames, called:

- "*General*": overall parameters ;
- "*Calibration*": calibration parameters (driven by the Eyelink host PC);
- "*Drift test position*": set the position of the drift prompt on screen;
- "*Binocular mode restriction*": choice of the recorded eye;

The "*General*" frame options are described in the following table:

LABEL	DESCRIPTION
Record Eyelink file (*.EDF, into EDF sub-folder)	If this option is ticked, data recorded in the "*Eyelink Data Viewer*" format (Eyelink host PC) will be transferred to the Eye and Pen PC.
Link port	For the time being, only the Ethernet link is supported.

The "*Calibration*" frame options are described in the following table:

LABEL	DESCRIPTION
Target diameter (pixels)	Calibration target diameter. Size of the "point" the participant has to look at for calibration.
Hole diameter (pixels)	Diameter of the "hole" in the middle of the calibration target.
Target color	Calibration target color.
Background color	Calibration window background color. This can be changed via the button shown to the right side. **Hint:** to enhance calibration, choose a background color similar to the one that will be used with the protocol displays.

The "*Drift test position*" frame allows you to set the horizontal and vertical position (in screen pixels) of the target shown on screen.

Clicking on the button selects the coordinates of the current screen center.

"Eye and Pen" only records data for one eye. The "*Binocular mode restriction*" frame option allows you to choose which eye (left or right) will be recorded.

b) ASL -504 (Applied Science Laboratory)

(*File/configuration/acquisition/Eye tracker* menu, "*ASL 504*" model, then "*Parameters*" button)
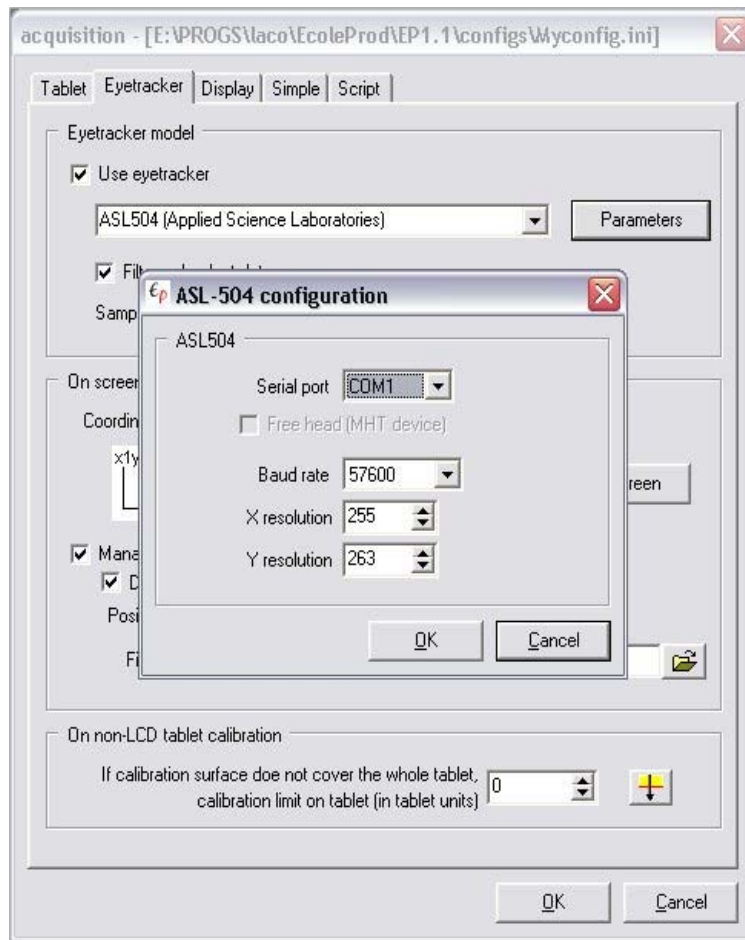
Figure 10: ASL-504 parameter configuration.

LABEL	DESCRIPTION
Serial port	Select your port from 9 possible COM ports.
Free head (MHT device)	If you want to use a head movement compensation device (if the eye tracker has one), tick this option.
Baud rate	Select the serial transmission speed.
X resolution	Select horizontal eye tracker resolution (consult the manufacturer's technical specifications).
Y resolution	Select vertical eye tracker resolution.

c) Eyeputer (Alphabio)

(*File/configuration/acquisition/Eye tracker* menu, "*EyePuter*" model, then "*Parameters*" button)
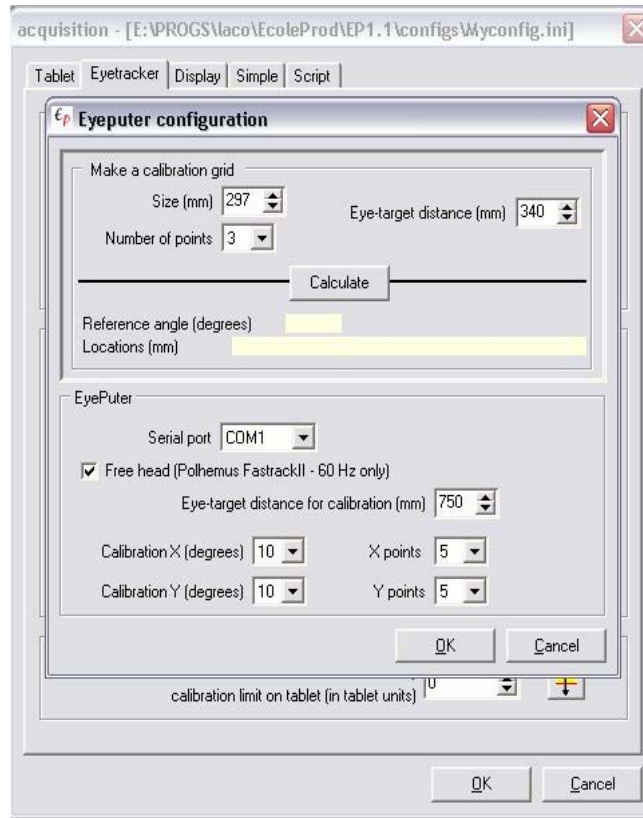
Figure 11: EyePuter parameter configuration.

The "EyePuter" configuration panel contains two frames, called:
- "*Make a calibration grid*": a tool to calculate distances between two points on the calibration grid;
- "*EyePuter*": EyePuter and calibration parameters.

"*EyePuter*" works in an angular system of coordinates. Accordingly, the "metric" distance between calibration points is not constant. Moreover, it does not have an on-screen calibration procedure.

To circumvent this problem, "Make a calibration grid" can help you to position calibration points on a sheet of paper that you can then place over the screen, etc., requiring calibration.

LABEL	DESCRIPTION
Size (mm)	Horizontal or vertical size of the calibration grid.
Number of points	Number of points used for calibration on the X or Y axis.
Eye-target distance (mm)	Distance between the participant's eye and the center of the calibration grid.
"Calculate" button	This button will calculate the angle between the calibration points and the metric distances between these points.

To create a calibration grid "by hand" on a sheet of paper, follow the steps below.

STEP	DESCRIPTION
1	Measure (with a ruler) the calibration grid length or width. Enter it (in millimeters) into the "*Size (mm)*" box.
2	Enter the distance between the participant's eye and the center of the calibration grid into the "Eye-target distance *(mm)*" box.
3	Select the number of calibration points you want for this dimension (horizontal or vertical) from the "*Number of points*" drop-down list.
4	Click on the "*Calculate*" button.
	The values in the "*Reference angle (degree)*" and "*Locations (mm)*" fields will be updated.
	"Re*ference angle (degree)*" indicates the true angle between two points.
	The values in "Locations (mm)" are the distance between the first calibration point and the following ones.
5	Using the values in "*Locations (mm)*", plot the points on the sheet.

The parameters for the "*EyePuter*" frame are described below.

LABEL	DESCRIPTION
Serial port	Select your serial (COM) port from the 4 possible ones.
Free head (Polhemus-FastrackII-60Hz only)	If you want to use a head movement compensation device (if your eye tracker is equipped with one), tick this box.
Eye-target distance for calibration (mm)	The distance between the eye and the calibration grid center. This option is only available if the "*Free head*" box is ticked.
Calibration X (degrees)	The theoretical angle between two calibration points on the horizontal axis.
Calibration Y (degrees)	The theoretical angle between two calibration points on the vertical axis.
X points	The number of calibration points on the horizontal axis.
Y points	The number of calibration points on the vertical axis.

(*File/configuration/acquisition* menu, "*Eye tracker*" tab, "*iViewX*" model, then "*Parameters*" button)
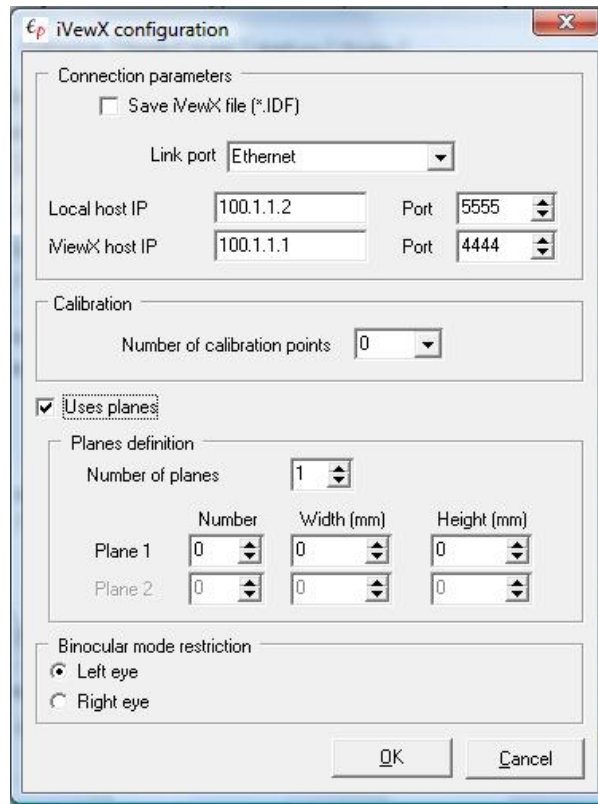
Figure 12: iViewX parameter configuration.

The configuration panel for "iViewX" contains four frames, entitled:

- "*Connection parameters*": settings to connect it to the eye tracker;

- "*Calibration*": calibration configuration;

- "*Planes definition*": parameters to set if your iViewX model records eye data relative to 3D planes (e.g. the HED-HT model);

- "*Binocular mode restriction*": choice of the recorded eye.

The "*Connection parameters*" frame allows you to determine how the Eye and Pen computer will "talk" to the iViewX computer. Options are described in the following table.

LABEL	DESCRIPTION
Save iViewX data file (*.IDF)	If this option is ticked, data will also be recorded on the iViewX host PC, in the "iViewX" format.
Link port	Only the Ethernet link is supported, for the time being.
Local host IP (and port)	Internet protocol address of the network adapter of the computer executing Eye and Pen. "Port" is the port number that will be used to communicate with the iViewX computer

iViewX host IP (and port)	Internet protocol address of the network adapter of the computer executing Eye and Pen. "Port" is the port number defined in the "Ethernet" configuration of the iViewX.

To achieve maximum communication efficiency, we suggest that you connect both computers' network adapters using an Ethernet crossover cable (recent computers may also use an ordinary cable), removing any other network connection, and using the following parameters (menu Start/Parameters/Network connections of Windows):

❖ "Eye and Pen" computer: IP address 100.1.1.2, subnet mask 255.255.255.0, neither gateway, nor DNS nor WINS.

❖ iViewX computer: IP address 100.1.1.1, subnet mask 255.255.255.0, neither gateway, nor DNS nor WINS.

In the iViewX software, select **Ethernet** in the "**Remote**" dialog box of the **Setup**-**Preferences**-**Hardware** menu. In the **Config** tab, select the Ethernet address and port. To allow Eye and Pen to send commands to iViewX, select the **Enable Data Streaming** option in the **Online** menu.

The "*Calibration*" frame options are described in the following table.

LABEL	DESCRIPTION
Number of calibration points	Number of points used for calibration You may choose 0, 1, 2, 5, 9 or 13 points. If you wish to let iViewX use its own calibration, choose 0 points

If your iViewX uses 3D planes (e.g. the HED-HT device), tick the "Use planes" checkbox and fill in the fields of the frame below.

The "*Planes definition*" frame allows you to configure parameters for each individual plane.

LABEL	DESCRIPTION
Number of planes	Number of planes (1 or 2) whose data will be read by Eye and Pen.
Plane 1 (number, width, height)	Number of the iViewX plane from which Eye and Pen is to read data, and its horizontal and vertical dimensions in millimeters.
Plane 2 (number, width, height)	Same as for plane 1.

If you are going to use two planes (as in the illustration below), several restrictions apply:

➢ Plane 1 must be located above plane 2;
➢ Both planes must be left-aligned;
➢ Planes can be of different widths;
➢ Planes 1 and 2 must lie in vertical proximity (contiguous).
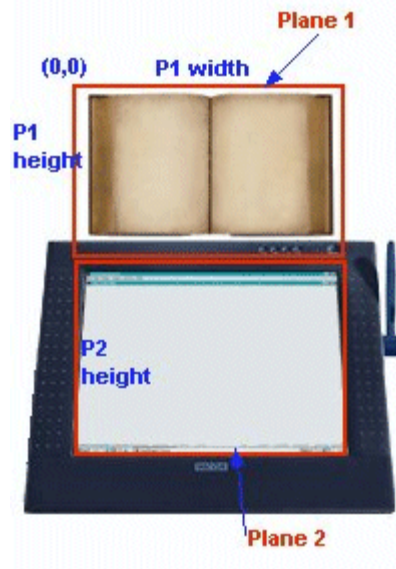
Figure 13: Setting up two planes.

"Eye and Pen" only records data for one eye. The "*Binocular mode restriction*" frame option allows you to choose which eye (left or right) will be recorded if your eye tracker is in binocular mode.

e) Tablet (pen tracking)

(*File/configuration/acquisition* menu, "*Eye tracker*" tab, *"Tablet" model*)

Strictly speaking, this is not an eye tracker, but rather a means of recording the movements of the pen across the tablet's surface as though it were eyetracking data. Thus, all pen movements within the tablet's detection area are recorded, even when the pen is raised ("in the air"). To a certain extent, this can help you to "see" where the pen goes during up pauses.

This "model" has no parameters to configure, so the "Parameters" button is inactive (shaded).

This "eye tracker"'s sampling rate is that of the tablet and its calibration onto the screen dimensions is matched to the display coordinates (Script tab).

Recording is matched as closely as possible to the requested sampling rate, whether or not there is new tablet data.

Note
When used in conjunction with the "masking" mode (cf. Display configuration), where the participant has to use the pen in order to be able to read the text on the screen (unmasking the display by moving the pen above the tablet's surface), you may end up with what amounts to the "poor man's eye tracker"!
When the pen moves out of the tablet detection area, data are fixed to -100,-100 pixels ("out of field").

II.2.3. Eye tracker calibration configuration

("*On screen calibration*" and "*On non-LCD tablet calibration*" frames)
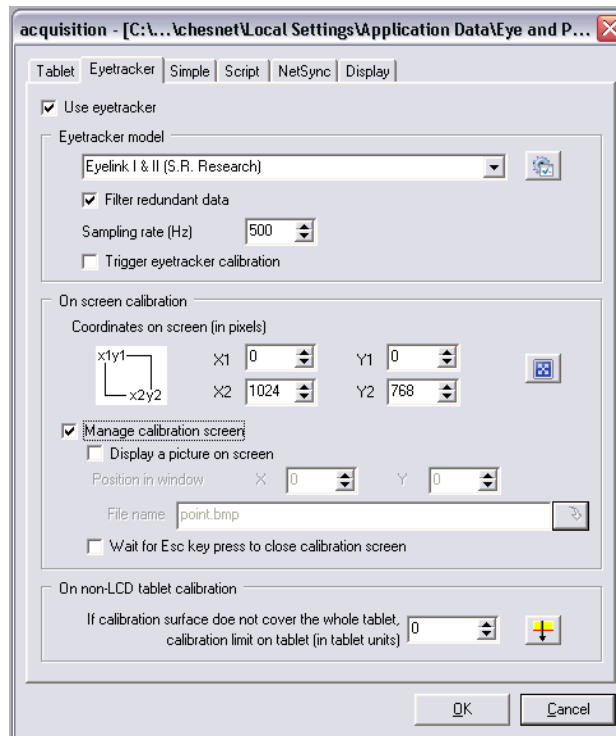
Figure 14: Acquisition configuration panel, "Eye tracker" tab.

a)   Calibration on screen

Details of the "*On screen calibration*" frame are given below:

LABEL	DESCRIPTION
Coordinates on screen (in pixels)	This option allows you to set the calibration grid coordinates. Click on the "**Whole screen**" button to use the entire screen surface.
Manage calibration screen	Set if Eye and Pen should provide a window (screen) for calibration (this is the case for Eyelink) or if the eye tracker manages the situation on his own.
Use a picture	Displays a background picture when calibrating (on screen). The following steps will help you to use this option. <table><tr><td>**STEP**</td><td>**DESCRIPTION**</td></tr><tr><td>1</td><td>Tick the "*Use a picture*" option.</td></tr><tr><td>2</td><td>Select the picture you want to be displayed during calibration (click on the "folder" icon). The path to this file is displayed to the right of the "**Filename**" label.</td></tr><tr><td>3</td><td>Enter the picture's upper left-hand corner horizontal coordinates (in pixels, relative to the display window) into the "**X**" box.</td></tr></table>

4	Enter the picture's upper left-hand corner vertical coordinates (in pixels, relative to the display window) into the "**Y**" box.
	The horizontal and vertical coordinates represent the location of the upper left-hand corner of the picture in the calibration window.

b) Calibration on non-LCD tablet

When calibrating on a non-LCD tablet (i.e. a tablet with an opaque surface, such as the one you can lay down on a desk), ask the participant to look sequentially at points drawn on a sheet of paper (laid on top of the tablet).
To learn how to calibrate your eye tracker correctly on the tablet, get in touch with your eye tracker reseller helpdesk.
Next, select the calibration limit on the tablet (this limit is zero by default).

Only leave the calibration limit at zero if you are using a **conventional tablet** and your calibration covers the tablet's entire surface.

If you are using a **tablet and a PC screen and want to record eye movements on both surfaces**, read the following section to find out how to define a calibration limit.

Why set a calibration limit?

Simply because using a non-LCD tablet and a computer screen has several limitations.
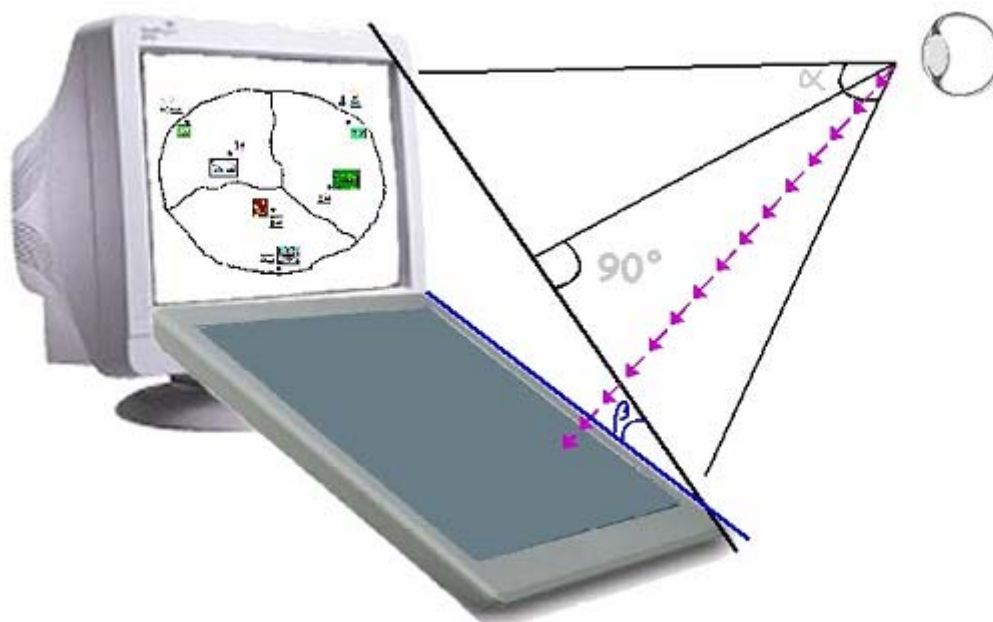
Figure 15: A tablet and a PC screen recording device.

On the one hand, the calibration area is restricted because most eye trackers are only able to make accurate recordings of eye positions within a 30° horizontal and 20° vertical range

(angle "α" on Fig. 15). Indeed, depending on the size of the screen and tablet, the calibration area may be assigned to only part of the screen and/or tablet.

On the other hand, because the tablet and the screen are not aligned in the same plane, the distance (depth) between each hardware device and the eye will vary. This variation can entail a decrease in recording accuracy. This is because eye trackers measure eye movements as though the watched scene was flat and perpendicular to the line of sight. With a device like the one shown in Figure 10, the eye-fixated point will be miss-estimated because of the angle between the tablet and the "theoretical" plane of vision (angle "β" on Fig. 15). In order to minimize the error, angle "β" has to be reduced as much as possible, by aligning the tablet and screen surfaces.

When the distance between the eye and the device prevents the participant from seeing the entire surface of the screen and the tablet, the calibration has to be restricted to the visible area (depending on the participant's vision and/or eye tracker's range). In this case, part of the task environment must be ignored. The vertical limit of this ignored area is referred to as the "**calibration limit**".
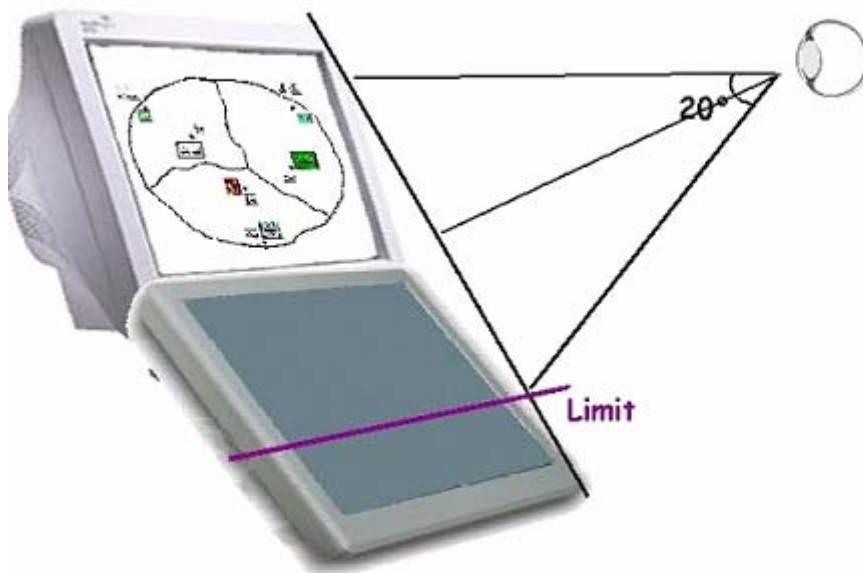
Figure 16: Calibration limit.

The steps to calculate the vertical calibration limit are described below.

STEP	DESCRIPTION
1	Click on the " ![icon] " icon to the right of the "*On non-LCD tablet calibration*" frame.
2	Touch the bottom of the calibration surface (calibration limit) with the pen (tablet stylus). The calibration limit value will be updated.
3	Click on the "*OK*" button. Mission accomplished.

III. DEVICE TESTS

Eye and Pen has functions that enable you to test the tablet and the eye tracker by two means:

- a visual test;
- a performance test.

The availability of the entries of the Device Test menu is determined by the acquisition devices being selected or not.

III.1. Tablet tests

III.1.1. Tablet visual test

(*Device tests/Tablet/Visual* menu)

This function allows you to check visually how the tablet is working within Eye and Pen.
A background picture or the background color is displayed (select from the *File/Configuration/Analysis* menu, "*Display*" tab).

Moving the pen across the tablet will leave a trace on the screen. The value defined in "*Configuration/Analysis/analysis/Draw a line between points*" is used to display the test. Either pen positions are « linked » with a line, or each pen position sampled by the tablet is plotted (no line between points), thereby giving a direct visual indication of data quality. When moving the pen across the tablet, pen pressure and coordinates (horizontal=X, vertical=Y) are displayed at the top of the screen.
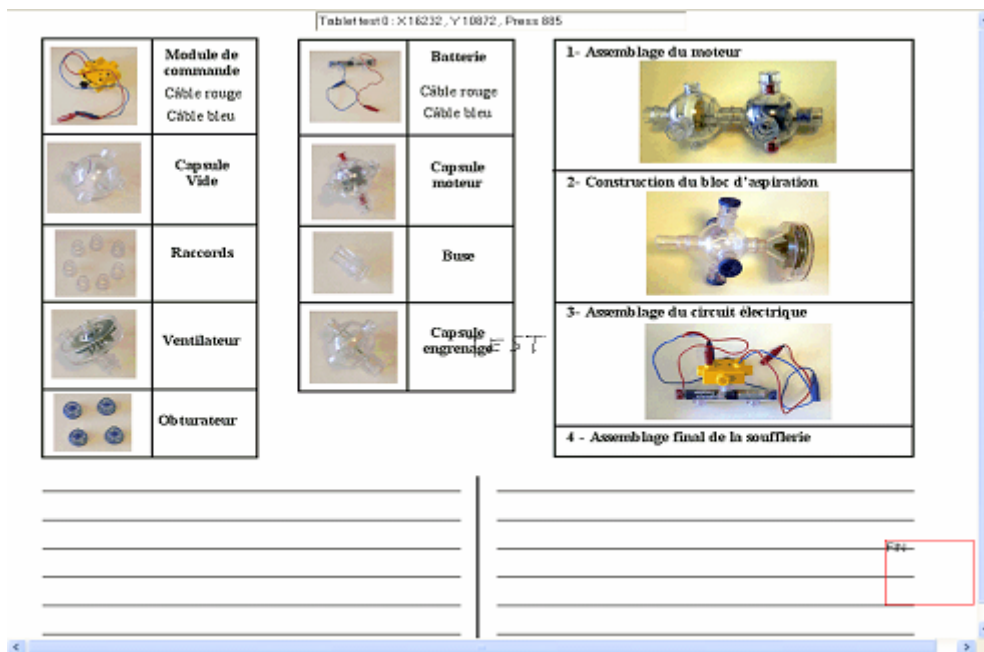
Figure 17: Visual tablet test.

To leave or end this test, you can either:

- press the "Escape" key;
- press the pen in the rectangle labeled "End" (see option "Show trigger zones" in [Analysis configuration](#) p. 48).

III.1.2.   Tablet performance test

(*Device tests/Tablet/Performance* menu)

The performance test simply consists in tracing a continuous line on the tablet. When you lift up the pen, an effective sampling rate is calculated.

Figure 18: a Tablet PC performance test.

This test may be used to determine a Tablet PC sampling rate.

III.2. Eye tracker tests

III.2.1.   Eye tracker visual test

(*Device tests/Eye tracker/Visual* menu)

This menu allows you to view the gaze position over the calibration grid (on screen).
It helps you to check the calibration visually.
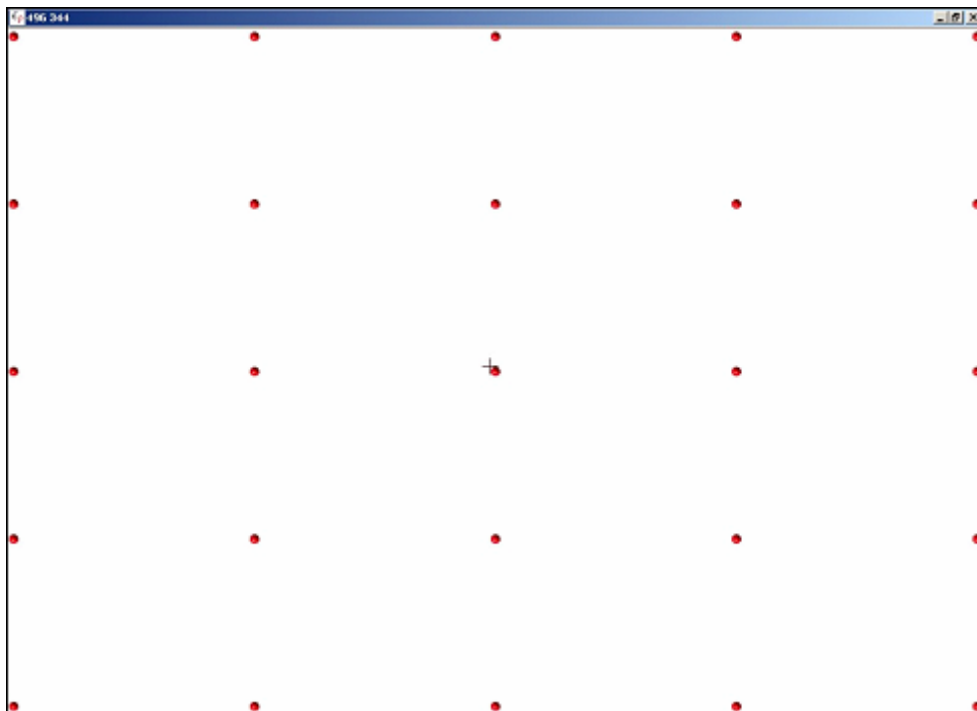
Figure 19: Visual eye tracker test.

Once you have calibrated your eye tracker, you should be able to see a cross (at the gaze position) "moving with your eyes" (center of figure), as it is measured by the eye tracker.
To cancel the test, press the "Escape" key.

III.2.2. Eye tracker performance test

(*Device tests/Eye tracker/Performance* menu)

This menu allows you to estimate the data sampling rate as recorded by "Eye & Pen".
When you launch this function, a message will be displayed at the top of the screen, asking you to wait for 15 seconds.

A message box will then display:

- the amount of data received per millisecond;
- the eye tracker's sampling rate.

Click on the "*OK*" button to end the test.

IV. DISPLAY CONFIGURATION PARAMETERS
(*File/Configuration/Acquisition* menu, "*Display*" tab)
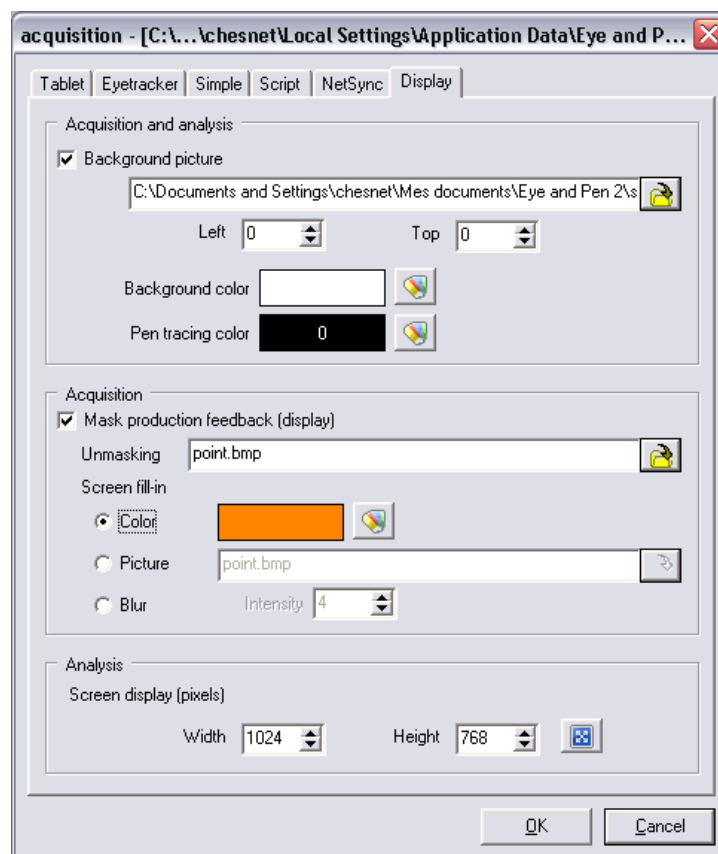
Figure 20: Acquisition configuration panel, "Display" tab.

The "*Display*" tab contains three frames:

- "*Acquisition and analysis*": displays defaults parameters common to the acquisition and analysis modules;

- "*Acquisition*": selects parameters specific to the Simple and Script modes of acquisition;

- "*Analysis*": displays parameters specific to the analysis.

LABEL	DESCRIPTION
Background picture	You can select a background picture by clicking on the folder icon. The path and name of this file will be displayed to the right of the label.
Left / Top	Allows setting coordinates (in pixels) of the picture's upper left corner, relative to the upper left corner of the display.
Background color	Select a background color for the display. **Note**: if you select a background picture that is as wide as the screen, the background color will not be visible.
Pen tracing color	Select the color of the trace left by the pen on the screen. A sample of the current color is shown to the left of the button. Its color number is written over the sample (useful for indicating a color in a script). **Caution**: if the tracing color is the same as the background color, it will not be visible.

The "*Acquisition*" frame described below allows you to set the acquisition only display parameters, be they Simple or Script.

LABEL	DESCRIPTION
Mask production feedback (display)	If you do not wish the screen content to be directly visible, check this option.
Unmasking	Select a picture file that will be used to "make visible" that which is not visible in the immediate vicinity of the pen ("hole" in the mask). An "umasking" file is a two-color (black and white) BMP format picture. The dark area represents the area around the pen location (the picture is centered on the pen's position). The white area represents the "peephole". In the following example, the unmasked part of the screen is elliptical and has been moved off-center to the left of the pen position, allowing the writer to see part of what has already been written. *Figure 21: A file used to unmask the display.* You will find several samples of files in the "Cursors" folder (the number in the filename indicates the "peephole" size in pixels).
Screen fill-in	Select the screen masking mode you require: • picture (select a file) • color • blur: the screen content is blurred. You can adjust the blurring intensity from 1 to 100%. Since the trace left by the pen is dynamically blurred, the higher the blurring level, the longer it takes to refresh the display.

Note: specific script commands allow a broader range of mask manipulations, such as the definition of locations to mask/unmask, etc. Also see the warning about masking, p. 69.

The "*Analysis*" frame allows you to configure display parameters that are only applicable for data analysis:

LABEL	DESCRIPTION
Width (pixel)	Width of the display window for later analysis.
Height (pixels)	Height of the display window for later analysis.
Current resolution ⊞	Directly fills "**Width**" and "**Height**" fields with the current screen display resolution (defined in the Windows display properties – see Windows Configuration Panel). **<u>Caution</u>**: the higher the resolution and color number, the longer it will take to "refresh" the display and the greater the video memory load will be.

V. ACQUISITION CONFIGURATION PARAMETERS

V.1. Simple acquisition

(*File/Configuration/Acquisition* menu, "*Simple*" tab)
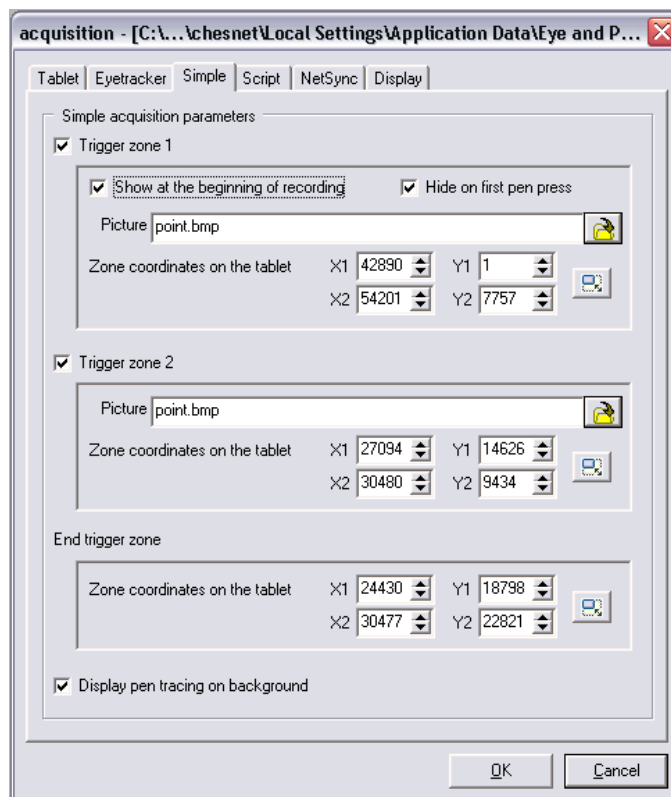
Figure 22: Acquisition configuration panel, "Simple" tab.

LABEL	DESCRIPTION
Trigger zone 1	This option creates a rectangular zone on the tablet, allowing you to display a picture on the screen. This will be referred to from now on as **Trigger zone 1.** In "simple acquisition" mode, if the participant presses the pen in a trigger zone, the selected picture is displayed on the screen. When the pen is released, the picture disappears. When it is ticked, this option allows you to select: • the trigger zone 1 coordinates on the tablet; • the associated picture; • the "*Show at the beginning of recording*" option; • the "*Hide on first pen press*" option.
Show at the beginning of recording	If this option is ticked, the picture "linked" to trigger zone 1 is displayed at the very start of data acquisition.
Hide on first pen press	This option is only available if the preceding option has been activated. If this option is ticked, the picture displayed at the start of recording is removed from screen when the pen is pressed on the tablet surface (i.e. typically when writing starts).
Picture (trigger zone 1)	Allows you to select the picture "linked" to trigger zone 1 (click on the folder icon). The name and path of the selected picture are updated in the box labeled "*Picture*". The picture is always displayed in the upper left-hand corner of the screen. According to its original size, the picture may cover only part of the screen (there is no resizing).
Trigger zone 2	Works in a similar way to trigger zone 1, except that it has no "Show at the beginning..." or "Hide on first..." options.
Picture (trigger zone 2)	Works in a similar way to the picture for trigger zone 1
End trigger zone	Allows you to create a rectangular zone on the tablet. When the participant presses the pen on it, the recording stops.
Zone coordinates on the tablet (in tablet lines)	Allows you to set tablet coordinates for trigger zones. Click on the icon.
Display pen tracing on background	If this option is ticked, the participant's writing (or drawing) will be displayed on the screen.

V.2. Script-based acquisition configuration

(*File/Configuration/Acquisition* menu, "*Script*" tab)
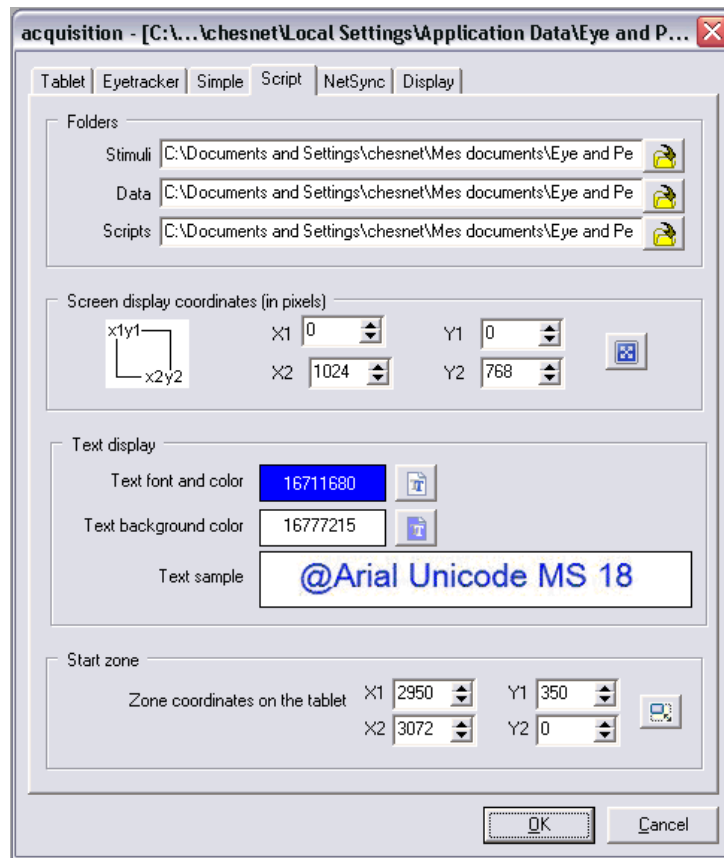
Figure 23: Acquisition configuration panel, "Script" tab.

The "*Script*" tab contains four frames:

- "*Folders*": names of the directories of recorded data and the stimuli used;
- "*Screen display coordinates (in pixels)*": parameters of screen display coordinates;
- "*Text display*": text display parameters;
- "*Start zone*": start zone coordinates on the tablet.

The "*Folders*" frame contains all the parameters needed to define the names of the directories containing data recordings and the stimuli that have been used.

LABEL	DESCRIPTION
Stimuli	Select the folder containing the stimuli used for script-based acquisition (pictures, movies, texts and / or sounds).
Data	Select the folder in which the data is to be recorded.
Scripts	Select the folder containing the script(s) to execute. When in a NetSync session, this folder's content will be announced (listed) to the Master host.

"*Screen display coordinates (in pixels)*" allows you to define which zone of the screen will be used to display stimuli, messages etc. To use the entire screen, set X1 and Y1 to zero, and set X2 and Y2 to the horizontal and vertical screen resolution values.

For example, with a 1024 x 768 resolution, set to 0, 0, 1024, 768 (see FAQ: "How can I find), or click on the button .

The "*Text display*" frame contains the parameters needed to define how texts will be displayed with a script.

LABEL	DESCRIPTION
Text font and color	Allows you to select default parameters for **text** and **message displays.** You can set: • color; • font; • size; • underline or strikeout style. When selecting a color (click on button), the new color and its color number are displayed to the left of the button. The color number is used in some script commands.
Text background color	Allows you to select the default background color for text displays. When changing the color (click on button), the color and its number are updated to the left of the button. The color number is used in some script commands.

The "*Start zone*" frame allows you to define a default start zone on the tablet.
When calling up the corresponding command in a script, the script will remain "on hold" until the participant presses the pen in this zone ().

V.3. NetSync: network-managed "Script" acquisition

(*File / Configuration / Acquisition* menu, "*NetSync*" tab)
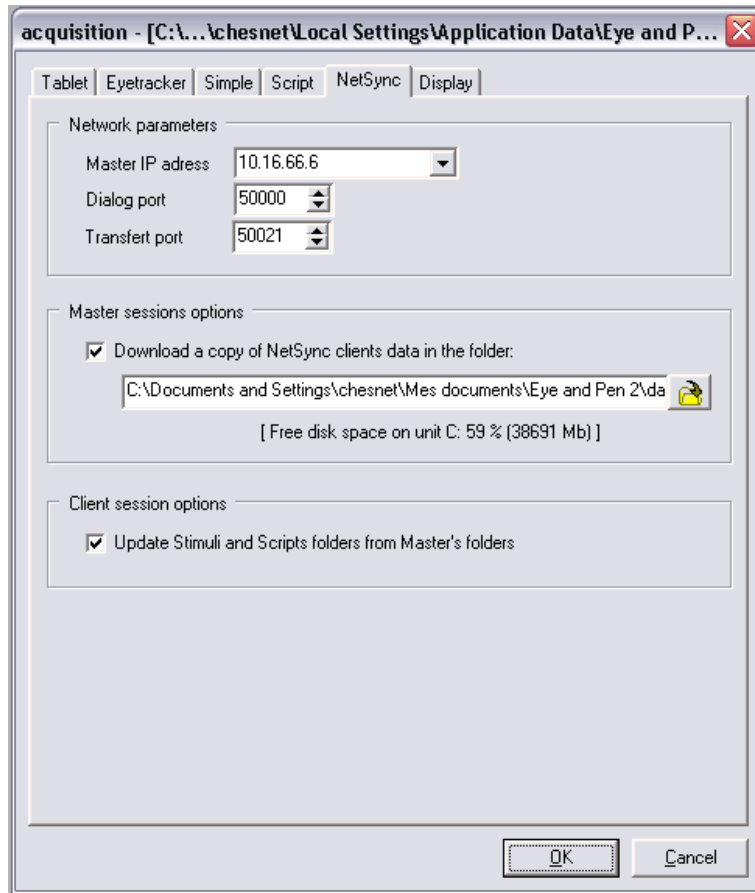
Figure 24: Acquisition configuration panel, "NetSync" tab.

The "*NetSync*" tab includes three frames:

- "*Network parameters*": parameters required for a "Client" host to be able to communicate with the "Master" host, and synchronize and transfer data;

- "*Master session options":* options used by the "Master" host.

- "*Client session options":* options to determine for a "Client" host.

The "*Network parameters*" frame contains all the parameters required to enable communication to take place between the Master host and the Client hosts. This information should be the same on both the Master host and the Client hosts.

LABEL	DESCRIPTION
Master IP address	Network address of the Master host. This address can be typed in. On the Master host, the scrolling list shows the host's local network address.
Dialog port	Port number used to manage communication (messages) between the Master host and the Client hosts.

Transfer port	Port number used to copy (FTP) data files to and from the Master host and the Client hosts.

All the available port numbers lie within the 50000-65535 range, i.e. the range authorized for "private" use on networks (see http://www.iana.org/assignments/port-numbers).

The "*Master session options*" frame contains parameters specific to the Master host. Thus, these options have no effect on client hosts.

LABEL	DESCRIPTION
Download a copy of NetSync client's data	Tick this option if you want the data collected during the acquisition session to be copied to the Master host. Next, click on the folder icon to designate the destination folder.

The "*client session options*" frame presents the option **Update Stimuli and Script folders from Master's folder.** Tick this option if you would like the *Script* folder (as defined in the *Script* tab) and the *Stimuli* folder to be updated with the content of the *Stimuli* and *Script* folders of the Master host. Only missing or older files are updated.

VI. DATA ANALYSIS CONFIGURATION PARAMETERS

VI.1. Analysis configuration panel

(*File/Configuration/Analysis* menu)

Once the data acquisition is complete, the data analysis tools will help you to sort, code, discard, cluster, etc. The options in this panel can be modified at any time.
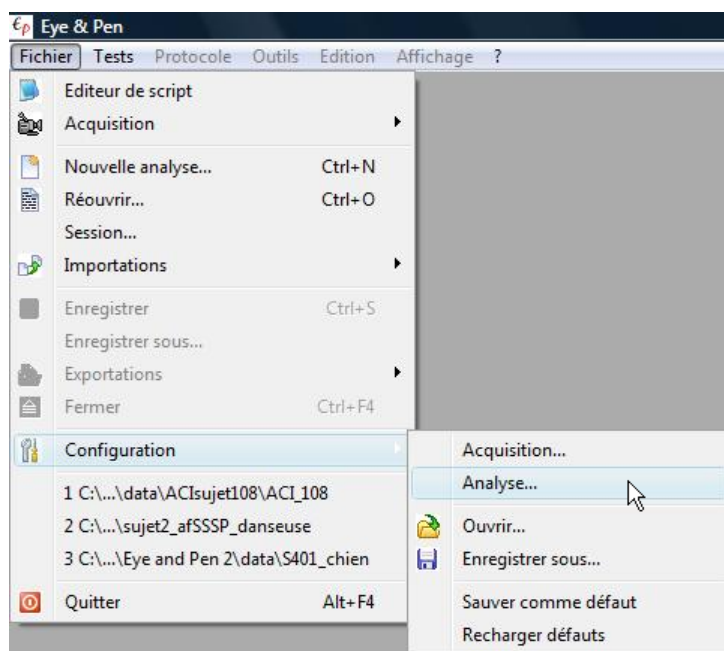
Figure 25: Launching the analysis configuration panel.

VI.2. New analysis configuration

(*File/Configuration/Analysis* menu, "*New Analysis*" tab*)*
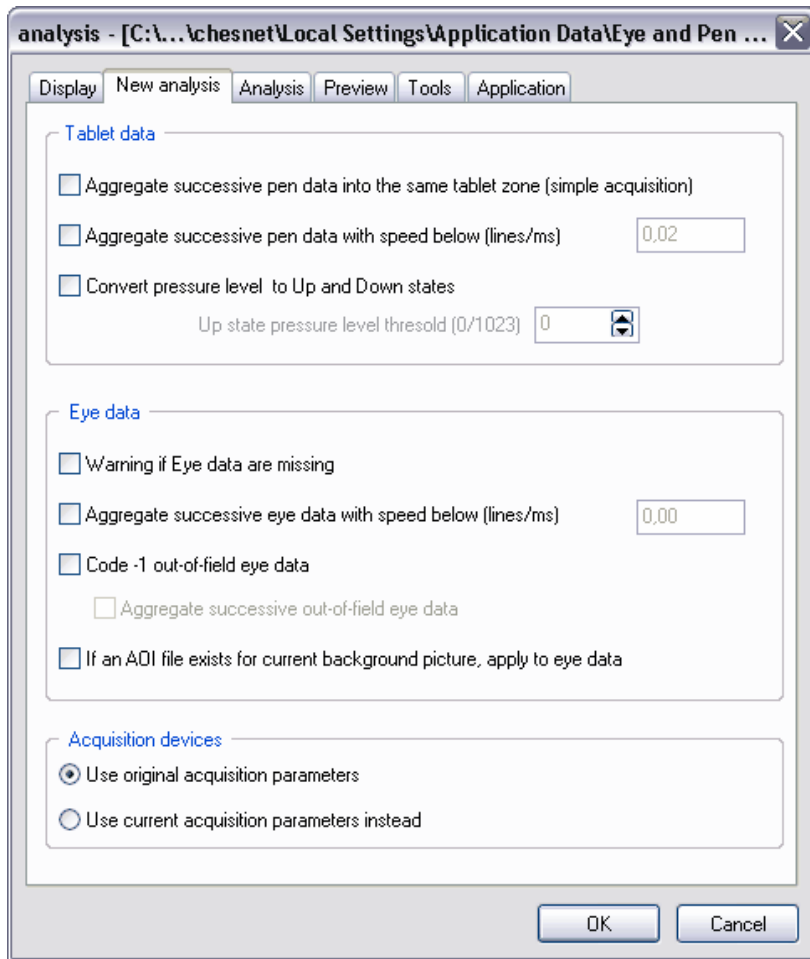
Figure 26: Analysis configuration panel, "New analysis" tab.

This tab shows three frames labeled:

- "*Tablet data*": parameters for tablet data treatment;
- "*Eye data*": parameters for eye tracking data;
- "*Acquisition devices*";

Parameters for the "*Tablet data*" frame options are explained below:

LABEL	DESCRIPTION
Aggregate successive pen data in the same tablet zone (simple acquisition)	If this option is ticked, successive data items in the same trigger zone will be turned into a single data item. All the participant's moves in this zone will be clustered within a single point.

	This option is useful when, for example, a participant moves slightly when pressing the pen in a trigger zone. A series of moves then becomes a single press.
Aggregate successive pen data with speed below… (lines/ms)	Aggregates successive pen data (pen being pressed on the tablet moves slightly) relative to a speed threshold. This threshold must be empirically determined. It is expressed in lines (tablet counting unit) per millisecond. All successive moves (samples) with a speed below this threshold will be clustered into a single point.
Convert pressure level to Up and Down states	Transforms pressure exerted by the pen on the tablet into a bi-level state: the pen is either "up" or "down". By default, a pressure level of zero corresponds to a pen "up" state. Any other value corresponds to a pen "down" state and is then set to 1023 (maximum pressure). This can be compared to the process of converting a picture in a monochrome series of grays into a black-and-white picture. **Caution**: the pressure scale may vary from one tablet model to another.
Up state pressure level (0/1023)	Allows you to define the threshold pressure value for the "up" state of the pen. For example, if the pen does not leave a trace on the paper until the pressure exceeds a value of 3 or 4, you can set the threshold to this value. **Important:** this option is only available if the "*Convert pressure level...*" option has been ticked.

The parameters for the "*Eye data*" frame options are described below:

LABEL	DESCRIPTION
Warning if eye data are missing	This option is only useful only if you are using an eye tracker. Otherwise, do not tick it. If you have ticked it, a warning message will be displayed if the protocol's eye-tracking data are not found. **Important:** if you are not using an eye tracker, do not tick this box. This option may be useful for signaling a data management problem, for example when only a part of a participant's files has been moved to a new directory.

Aggregate successive eye data with speed below (lines/ms)	Aggregates successive eye data below a certain speed threshold. This threshold must be empirically determined. It is expressed in lines (tablet counting unit) per millisecond. All successive moves (samples) with a speed below this threshold will be clustered into a single point.
Code -1 out-of-field eye data	Data with coordinates beyond the calibrated area are automatically coded as -1.
Aggregate successive out-of-field eye data	This option is only available if the above option has been ticked. This option allows you to aggregate (cluster) successive data beyond the calibrated area. These items will not be taken into account in the analysis, but will enhance the initial data duration with the sum of all subsequent discarded data. It is useful to reduce the amount of data when, for example, the participant is thinking with his/her eyes "gazing skywards".
If an AOI file exists for current background picture, apply to eye data	If an AOI[1] file exists with the same name and in the same directory as the background picture (e.g. "backgnd.bmp" and "backgnd.aoi"), this option allows you to automatically load and apply AOI zone coding to the eye data.

The "*Acquisition devices*" frame allows you, either to use the original device(s) configuration parameters that were in use when the protocol was recorded, or to use the current devices configuration instead (which may not be the same).
Since these parameters includes devices resolutions and sizes, this will have consequences on data display and later calculations (you can see which parameters are included in file headers in Appendix).

As a rule, it is advisable to use the original configuration parameters.

---

[1] An AOI is a visual Area Of Interest (see p.140)

VI.3. Analysis configuration

(*File/Configuration/Analysis* menu, "*Analysis*" tab*)*
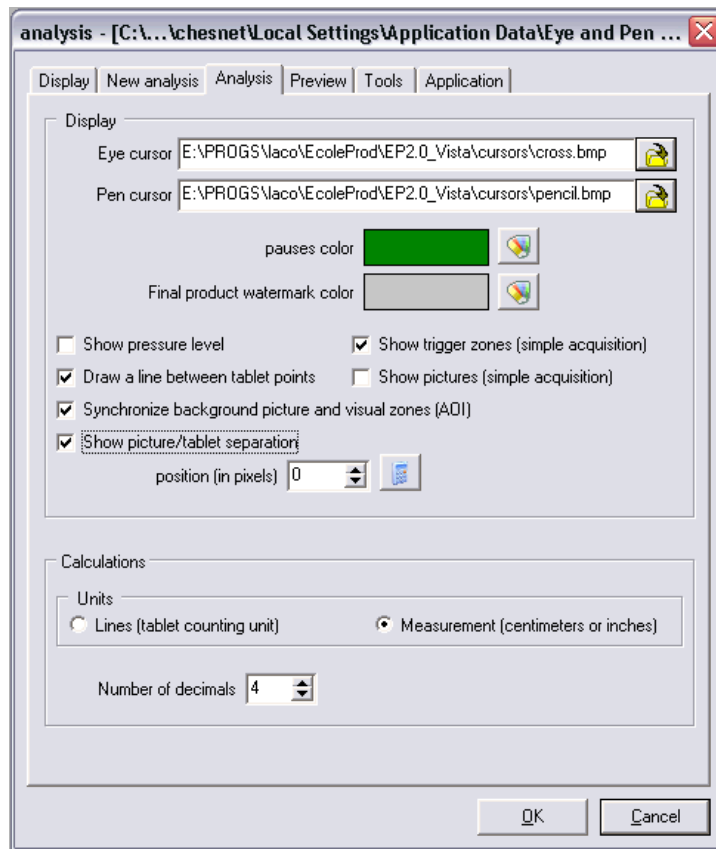
Figure 27: Analysis configuration panel, "Analysis" tab.
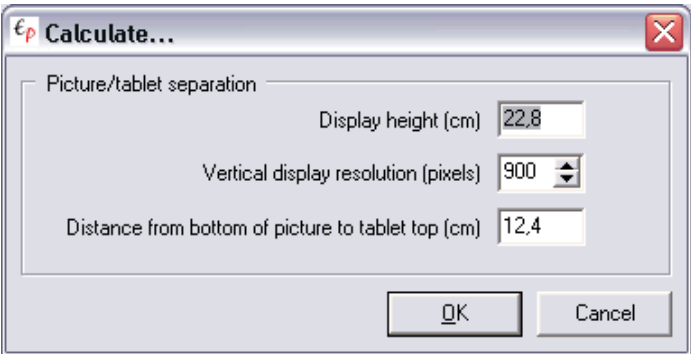
This tab shows three frames, labeled:

- "*Display*": parameters for data display options;
- "*Calculation units*": parameters for the calculation results;

The "*Display*" frame options are described below.

LABEL	DESCRIPTION
Eye cursor	Choose the icon that will materialize (show) gaze position on the screen. A cursor is a ".BMP" picture file, with a maximum of 256 colors. The first point (pixel) in the lower left-hand corner of the picture defines the color that will be replaced by transparency. For example, you can create a cursor with a circle shape corresponding to the participant's central vision, or fovea (consult fovea.bmp in the \cursors subdirectory to gain a better idea of what is "centrally" viewed.
Pen cursor	Choose the icon representing the pen's position on the screen.
Pause color	Allows you to change the color used to represent pauses on the screen (see p.116).

(button image)	Click on the button, select a color and the color sample will be updated to the left of the button.
Final product watermark color (button image)	This option allows you to see what the participant wrote as a background. The final product is then displayed as a watermark. Clicking on the button, you may then choose a particular color for the watermark. A sample of the selected color is displayed to the left of the button. Hint: to make the watermark disappear, choose a color that is similar to the background color.
Show pressure level	Change the color of the pen's trace according to the amount of pressure exerted on the tablet (shades of gray). The greater the pressure, the darker the shade (0=white, 1023=black).
Draw a line between tablet points	This option allows you to draw a line joining up successive points. It makes the participant's writing easier to read. Unticking this option makes it easier to assess data quality in an initial analysis (number of points, etc.). The distance between the points will be related to the participants' writing (drawing) speed.
Synchronize background picture and visual zones (AOI)	Visual Areas Of Interest (AOI) are commonly defined with respect to the background picture. If the background picture is moved (cf. Shift layers, p.136), the AOI will no longer be in the right place. This option allows you to "link" the AOIs to the background picture. That way, if the background is moved, the AOI will move with it.
Show picture/tablet separation	Check this option if you are using a non-LCD tablet + screen (the tablet is located below the screen) and you are calibrating your eye tracker on the whole device (cf. diagram, p. 33) Displays a separation line between screen and tablet.
Position (in pixels) (button image)	Determine the position of the separation line, taking into account the physical position of the two devices. Click on the button, fill in the fields of the dialog box and click on "*OK*". *Figure 28: Physical separation between screen and tablet dialog box.*

Show trigger zones (simple acquisition)	If this option is ticked, trigger zones will be displayed in the background during analysis. In general, it is advisable to keep this option ticked.
Show pictures (simple acquisition)	If this option is ticked, the picture linked to a trigger zone will be displayed when the pen data "enters" into this zone. It mimics the participant's behavior.

Parameters in the "*calculations*" frame are described below:

LABEL	DESCRIPTION
Units	This frame allows you to select the unit of measurement for the calculation outputs. There are two possible values: • in lines, the tablet's "natural" unit of measurement; • in centimeters or inches, depending on the tablet's resolution and the driver's measurement system.
Nunber of decimals	Set the number of decimals for non integer values extracted (distances, speed, etc.).

VI.4. Analysis display configuration

(*File/Configuration/Analysis* menu, "*Display*" tab*)*
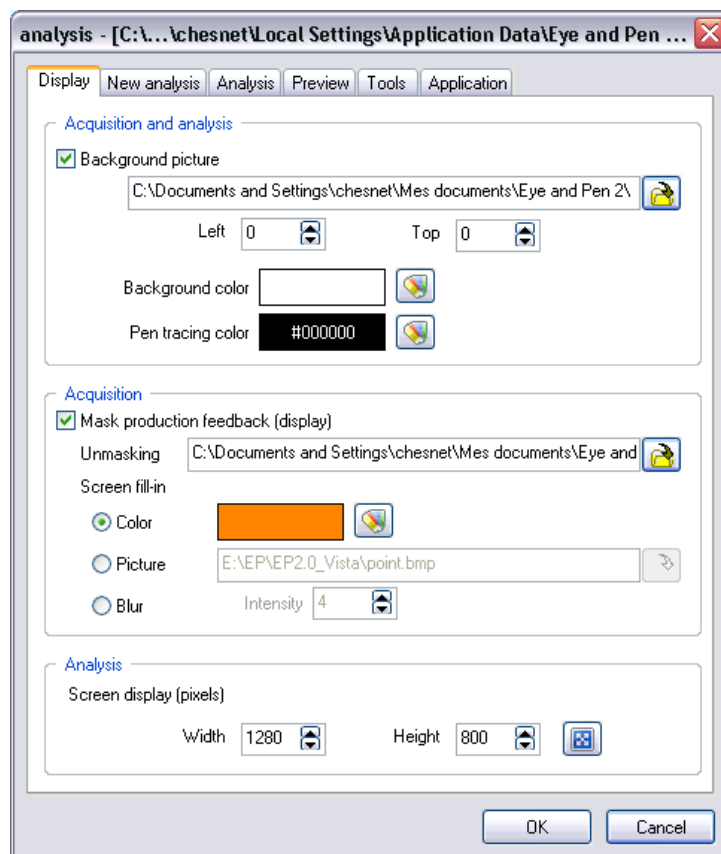
Figure 29: Analysis configuration panel, "Display" tab.

This tab is the same as the one in the acquisition configuration panel (File/Configuration/Acquisition/<u>Display</u> menu, p. 37)

VII. PREVIEW CONFIGURATION

(*File/Configuration/Analysis* menu, "*Preview*" tab*)*
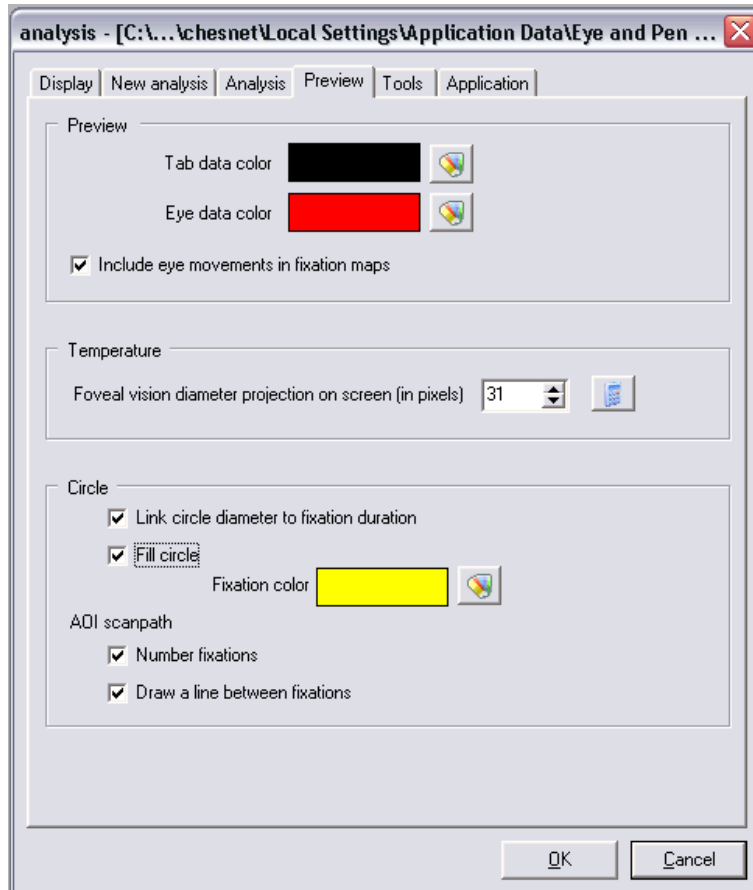
Figure 30: Analysis configuration panel, "Preview" tab.

The "*Preview*" tab shows three frames, labeled:

- "*Preview*": set colors to represent data projection on screen;
- "*Temperature*": parameters specific to this preview mode;
- "*Circle*": parameters specific to this preview mode.

The "*Preview*" frame allows you to set:

Tab data color	Select a color to represent tablet data (click on the button). A sample of this color will be shown to the left of the button
Eye data color	Select a color to represent eye data (click on the button). A sample of this color will be shown to the left of the button.
Include eye movements	If this option is ticked, previews like "temperature" will

in fixation map	include all eye movements' data, whether it is fixations or simple eye movements.

The "*Temperature*" frame allows you to specify **the foveal vision diameter projection on screen** (in pixels). To set this value, click on the "*Calculate*" button. The following dialog box will then be displayed:
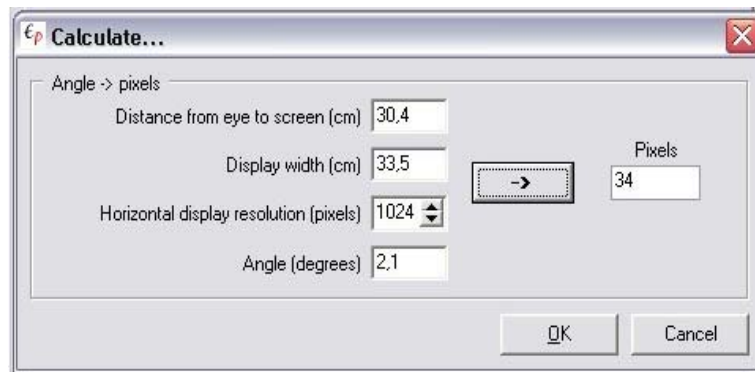
Figure 31: Calculation of foveal vision diameter on screen.

Fill the boxes with the requested values and click on the "->" button. When satisfied, click on the "*OK*" to validate.
The projection of the foveal vision will then be calculated and displayed to the left of the "*Calculate"* button*.

The "*Circle*" frame options are described below:

Fill circle	Should the circles representing fixations be filled with color? If the answer is "yes", tick this box. Then, click on "*Change*" to select the color. The current color is shown to the left of the button.
Link circle diameter to fixation duration	If this option is ticked, the diameter of the circle representing a fixation will grow according to fixation duration. The longer the fixation, the greater the diameter. [diameter = $(Log_2(Duration) * log_{10}(Duration)-1) / 2$].

AOI scanpath: successive fixations in the same AOI are summed up and the "point" is plotted at summed fixations barycenter location coordinates. Obviously, if you want this preview to be meaningful, fixations (see p. 126) and AOI (see p. 140) should have been calculated and defined first.

Number fixations	Fixations are numbered in chronological order, which allows you to make their order visible.
Draw a line between fixations	Each circle, representing a fixation, is linked to the next one with a line. This visualization may help to follow a scan path.

VIII. CONFIGURING TOOLS

(*File / Configuration / Analysis* menu, "*Tools*" tab)
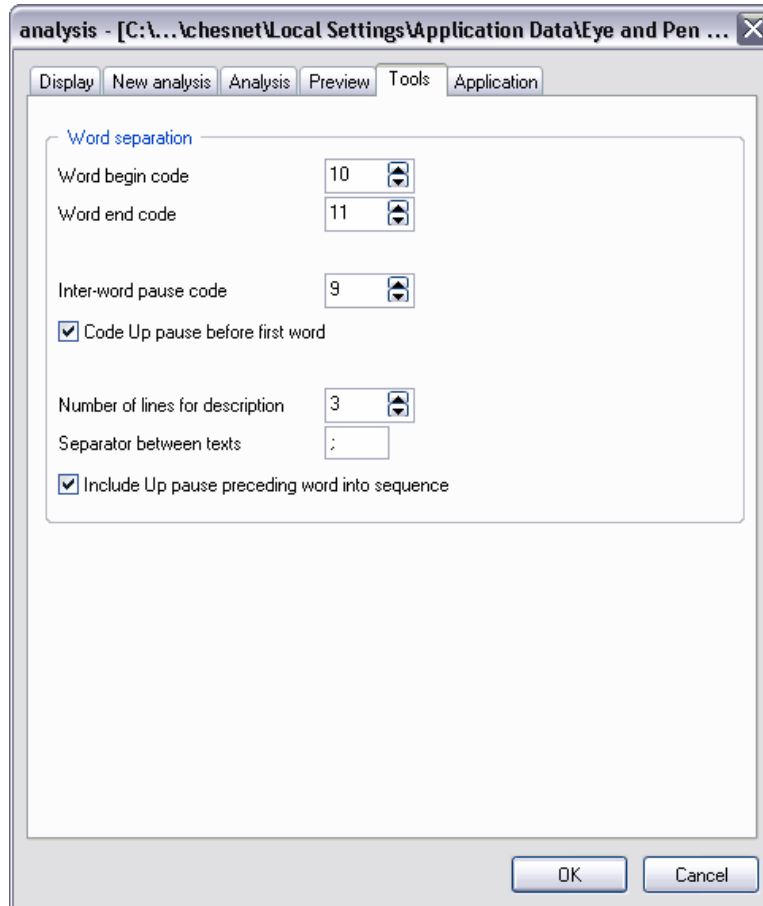
Figure 32: Tools parameters.

The "*Tools*" tab includes the options of the Tools menu items.

The "*Word separation*" frame allows you to set the default values for the tool of the same name.

The options of the "*Word separation*" frame are described below:

LABEL	DESCRIPTION
Word begin code	Code that will be attributed to the first tablet datum for each word. This code can be modified for each individual word when performing step 2.
Word end code	Code that will be attributed to the last tablet datum for each word. Once again, this code can be modified when performing step 2.
Inter-word pause code	Code attributed to the pause(s) preceding each word.
Code Up pause before first word	This option allows you to determine whether the pause before the first word is to be coded or not. In most cases, this pause is the pre-writing pause, so you may want to exclude it from

	the automatic coding.
Number of lines for description	Number of text lines available to describe each separated word (from 1 to 6).
Separator between texts	You may decide to insert separators between the lines of description for each word ("Text" lines). In this case, the description lines for each word will be assembled together with the separators to create a single descriptor that can be used to generate sequences.
Include Up pause preceding word into sequence	If this option is ticked, the pause preceding the word will be included in the sequence as its start timestamp.

Word separation data are automatically saved / loaded with the protocol. The file has a "WDS" extension (*<ParticipantName>*.WDS) and is saved in the same folder as the analysis data file. It is automatically loaded (if it does exist) whenever the analysis is reopened. This WDS file can be manually deleted (without any risk for Eye and Pen).

IX.   APPLICATION PARAMETER CONFIGURATION

(*File / Configuration / Analysis* menu, "*Application*" tab*)*
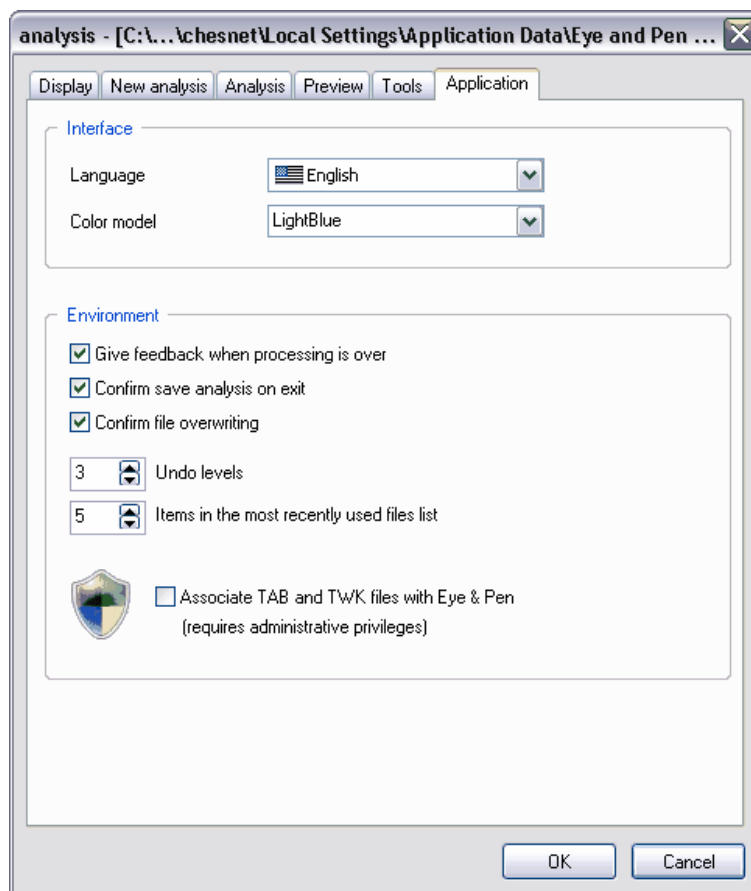
Figure 33: Application parameter settings.

The "*Application*" tab includes two frames:

- "*Interface*": select language and toolbar appearance;
- "*Environment*": general software settings;

The options of the "*Interface*" frame are explained below:

LABEL	DESCRIPTION
Language	This option allows you to select the software interface language in a scrolling list (English 🇺🇸 or French 🇫🇷).
Color model	This scrolling list allows you to select a "color model" to personalize the toolbars look and feel (Pauses, Fixations, Sequences, etc.). Note : This option uses a special folder named "*Toolbars*", containing the color models. Each model is made of a sub-folder (named after the color model name) containing pictures and icons for toolbars (as BMP format pictures), and a file named "Toolbars.ini" describing the background color and type font involved. The "*Toolbars*" folder contains a file named "ColorModelsHowTo.txt" explaining how to build your own color models.

The "*Environment*" frame allows you to set specific behaviors:

Give a feedback when a processing is over	If this option is ticked a message will be displayed when some processing ends, for example capturing the protocol into a picture or clustering eye movements into fixations.
Confirm save on exit	If this option is not ticked, the analysis will automatically be saved, without any prompt. Otherwise, you will be asked to decide whether you want the analysis to be saved or not.
Confirm file overwriting	Allows you to activate / deactivate the overwrite confirmation prompt messages when attempting to save over an already existing file. Note : Although the warning speaks about file replacement, when it come to text files (data extraction, etc.), data are appended to the existing text file. Deactivate this option may be useful in particular cases, for example to simplify data extraction automation. In the usual case, it is advisable and safer to leave this option ticked.

Undo levels	Set the number of successive actions that can be cancelled.
Items in the most recently used files list	Set the maximum number of filenames (saved analysis) displayed at the bottom of the File menu when scrolling down. Admitted values ranges from 0 to 128.
Associate TAB and TWK files with "Eye and Pen" (Windows registry)	Allows you to directly launch "Eye and Pen" when you click on a ".TAB" data file or a ".twk" analysis file (details in <u>Appendix XIII</u>).

X.   SAVE / LOAD CONFIGURATION PARAMETERS

When you use "Eye and Pen", the last parameters to have been used are reloaded by default.

If you wish to use more than one configuration, you can **save** current parameters under a new name before modifying them. You will then be able to reload them later.

To do this, click on the "*File*" menu, select "*Configuration*" and click on "*Save as*". The following dialog box will then be displayed*:*
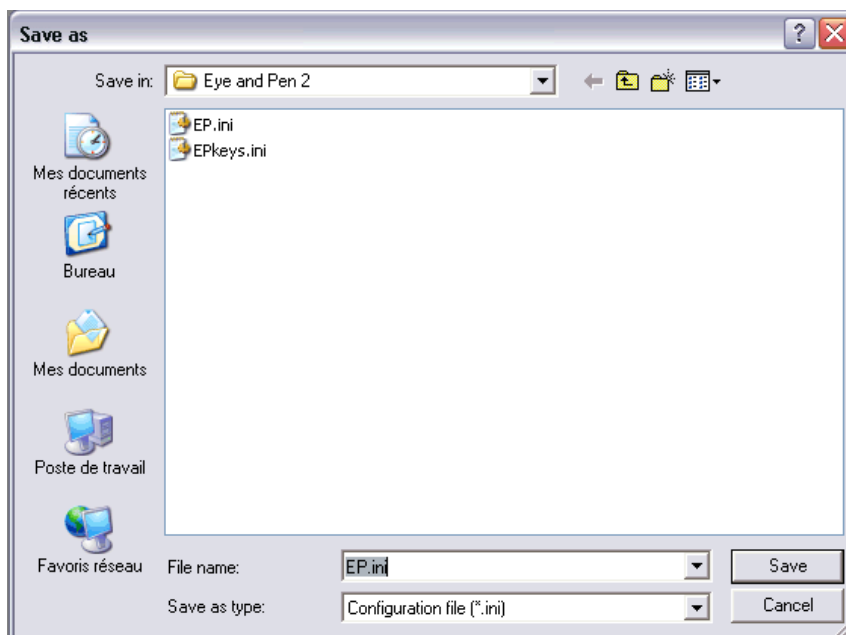
Figure 34: Saving the configuration parameters.

Choose a filename and a directory and save your configuration parameters (".ini" file type) clicking on the "*Save*" button.
The file has now been saved.

To **open** a parameter file, click on "*File",* select "*Configuration*" and click on "*Open*". The following dialog box will then be displayed:
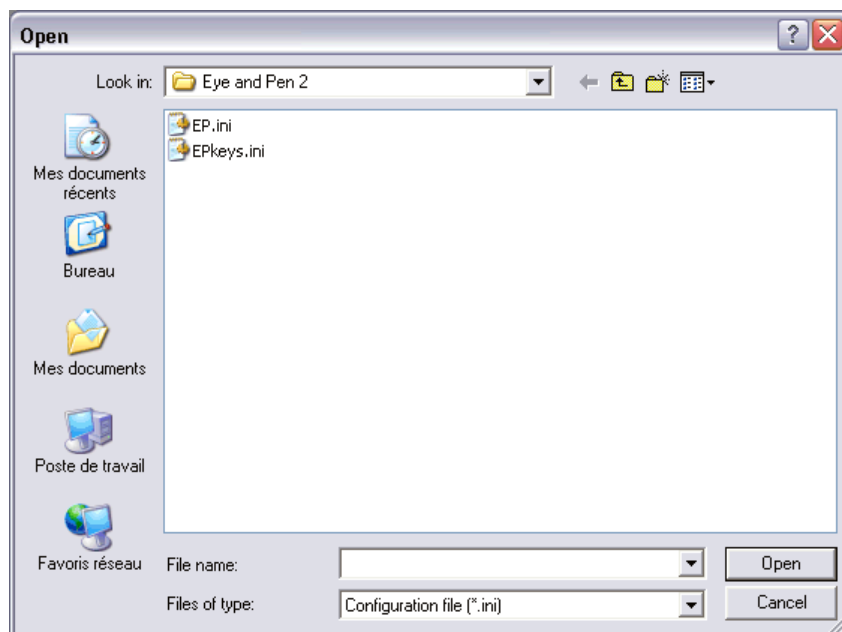
Figure 35: Loading the configuration parameters.

Select an ".ini" configuration file and click on "*Open*".

The configuration parameters contained in this file will immediately be activated.

Default configuration

If necessary (e.g. you misconfigured some parameters), you can **reload** the Eye and Pen **default parameters**.
To do so, click on "*File",* select "*Configuration*" and click on "*Reload defaults*".
Default configuration parameters will now be activated.

To save **current parameters as new default parameters** for Eye and Pen, click on "*File",* select "*Configuration*" and click on "*Save as default*".

Note:

You cannot load a configuration file or the default parameters once an analysis is underway. Otherwise, the parameters of the tablet, eye tracker and display, together with other parameters involved in the analysis might be modified and your analysis would be greatly compromised.
For this reason, if you wish to change the configuration file, close your analysis first.

CHAPTER 2: data acquisition

I. INTRODUCTION

I.1. Acquisition modes

Data acquisition can be managed in tree modes:

- a so-called "Simple" acquisition mode, which is totally defined in the Eye & Pen configuration panel;

- an acquisition mode driven by a "Script", i.e. where all events are described through a pseudo command language. Certain basic parameters are set in the Eye & Pen configuration panel, but all the parameters may be redefined via the script.

- an acquisition mode driven by a "Script" and synchronized through a network: NetSync.

The "**Simple**" mode allows you to manage an acquisition according to the options selected in the configuration panel. For example, should a picture be displayed when the recording starts, should this picture be hidden as soon as the participant begins to write, etc.
The ease with which Simple acquisition can be managed brings with it certain limitations, in that the options are limited to what is available in the Eye and Pen configuration panel.

So that you can run a more complex (or extended/varied, etc.) acquisition scenario, Eye and Pen also provides a "**Script**" mode.

A script is based on a programming mini-language that allows you to specify how the acquisition will take place and what will happen[2]. In short, it describes the acquisition scenario that Eye and Pen will have to process.

In practical terms, a script comprises a text file containing written commands and parameters that Eye and Pen will interpret and then execute: for example to play an audio stimulus, to wait until the pen is pressed in a tablet zone, to load a list of words to display on the screen, etc.

---

[2] A script language is a programming language that allows control of functionalities of a computing system *[...]*. Script language can free from low level constraints *[...]* and benefit from a high level syntax.
A script language is most often executed from files containing the source code of the program that will be interpreted. Historically, they were created to shorten the traditional dveleopment process *[...]*. The first languages were oftenly called "command language" or "Job Control they allow you to automate a bunch of successive simple commands, like a "theater script" (translated from http://fr.wikipedia.org/wiki/Langage_de_script).

I.2.   Session logging

During a recording session, the main events of interaction with the participant (on-screen displays, audio/video playing, recordings, responses, etc.) are recorded in a ".log" text file, together with the time at which these events occur (time elapsed in milliseconds since the beginning of the recording session).

This Log may be used again later, when analyzing data.

I.3.   About timing

Tablet and eye-tracking data are recorded with a common time base, i.e. the time at which the PC executing "Eye and Pen" starts receiving the incoming data.

Although this system is not ideal (timing may be less reliable than that of the acquisition device because the data have to "travel through" the Windows system, resulting in additional and unpredictable delays), this mode was chosen to circumvent problems encountered with some tablet drivers. Moreover, not all eye trackers time-stamp their data.

I.4.   Acquisition environment

The tablet and eye tracker that you have selected and configured are activated as soon as data acquisition begins.

If the eye tracking system is capable of doing so, its calibration procedure is automatically and immediately launched at the start of data recording, without any request on your part.

For EyePuter and ASL504 (serial mode), you will have to carry out an independent calibration prior to acquisition.

The acquisition is monocular, i.e. only the right or left eye (cf. Eyelink configuration panel, p. 24).

Abort an acquisition: whatever your acquisition mode, a key press on the **F12** keyboard key will halt the acquisition as quickly as possible.

II. "SIMPLE" ACQUISITION

(*File/Acquisition/Simple* menu)

The "Simple" mode of acquisition allows you to record tablet and eye data (if you are actually using an eye tracker, of course) using the simple acquisition parameters defined in the acquisition configuration panel.

Steps to launch the Simple acquisition mode are described below.

A dialog box will appear, asking for the name of the file in which to save the recorded data.

STEP	DESCRIPTION
1	Select the directory in which you want the data to be recorded.
2	Give the file a name, such as the participant's name (example: participant 1).
3	Click on the "*Save*" button. The "*Go*" button will then become available.
4	Click on the "*Go*" button to start your protocol and data recording.

To end the session using the simple acquisition mode, you have two solutions:
- press the pen in the "end" trigger zone on the tablet (normal exit);
- press the "*Escape*" key ("emergency" exit).

III. SCRIPT-BASED ACQUISITION

Acquisition based on script relies on a mini programming language.
This pseudo language is made up of a list of **commands.**
These commands have to be written in a **text-only file ".txt"** (Windows ANSI coding).
There must be only one command per line. No spaces (blanks) are allowed.
A set of commands in a text file is called a script.

In a script, a line may contain three sorts of items:
- a **command**
- a **comment**: defined by a semi-colon at the beginning of the line and followed by text (spaces allowed)
- a **label:** defined by "**:**" followed by a single word.

Example:

Command1
; my comment is here ⟵——————— A comment
command2
:Tag1 ⟵——————— A label
Command3

III.1. Script-based acquisition dialog box

(*File/Acquisition/Script* menu)

When launching the script-based acquisition mode, a "start" dialog box is displayed on the screen.
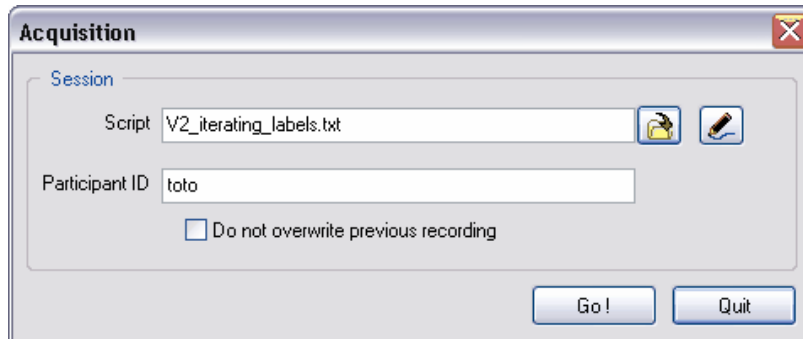
Figure 36: Script-based acquisition dialog box.

LABEL	DESCRIPTION
Script	Displays the name of the current script.
	Allows you to select the script you want to use. **Caution:** a script file must be saved in "text-only" file format (".txt") if it is to be understood by Eye and Pen. By default, the last used script is proposed.
	Modifies the current script or create a new one in the script editor.
Participant ID	Choose a name for the data file(s).
Do not overwrite previous recording	This option allows you to avoid losing data from a previous recording if you mistakenly choose the same name. If this option is ticked and a data file with the same name already exists in the same directory, a warning will be displayed (recording is canceled).
Go !	Launches acquisition. First, the script is checked against major defects (syntax faults, missing files), then it is executed.
Quit	Closes the acquisition dialog box and returns to the main Eye and Pen screen.

III.2. Script editor

(*File / Script editor* Menu)

The script editor helps you to write commands by selecting values and filling in fields in dialog boxes. If a script file is named in the "Script" field of the script acquisition panel, it can be modified. If the panel does not already contain a name, a new file will be created.
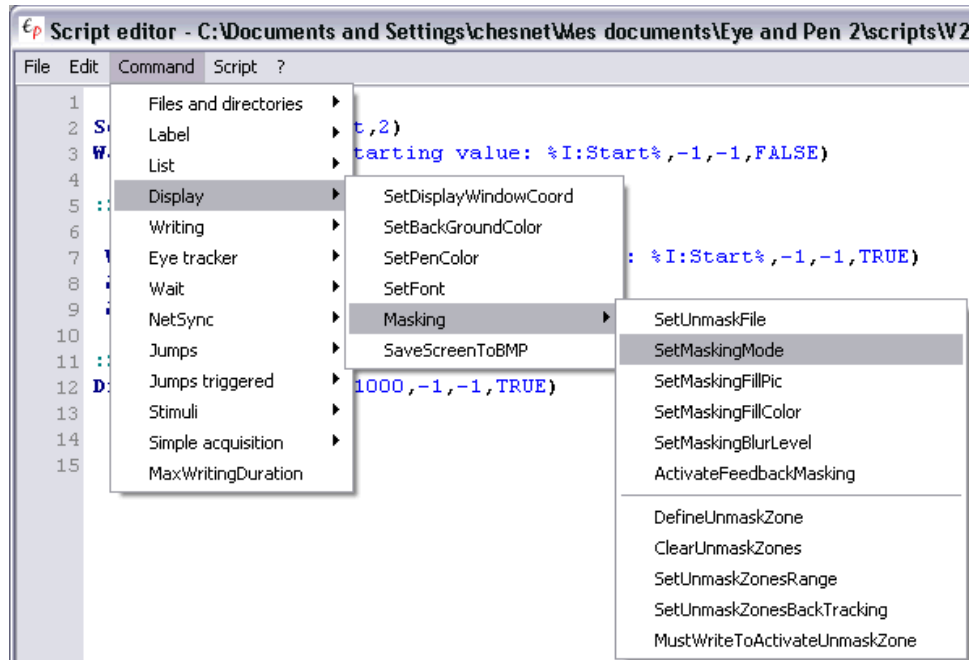
Figure 37: Script editor menus.

The "*Command*" menu allows you to select script commands and set their parameters with the help of dialog boxes where you fill in fields. Obviously, a script can also be written or modified "by hand".

Depending on the type of zone/area to be defined, selection will be done either with the computer mouse on the screen or else with the tablet pencil on the tablet. For some fields, a mouse right-click will open a list of suggestions, including labels found in the script, keywords and keyword combinations.

When it has been validated, the dialog box will write the command for you at the place where the text cursor is, with the values you have defined.

Commands are grouped by category and presented in the following order

The "*Script*" menu:
- "*Check syntax*" performs an automatic verification to ensure that the script is "well" written.
- "*Execution test*" checks that all the elements involved in the script (syntax, stimuli files, etc.) are ready.

The " ? " menu:
- "*Help*": shows what help is available for script acquisition
- "*Command list*": provides a list of script commands

Once the script has been saved, its name is updated in the script acquisition panel.

III.3. Script commands.

Script uses two kinds of commands:

- commands comprising a single word.
    Example: `Command1`

- commands comprising a command word followed by parameters between brackets, separated by a comma.
    Example: `Command1(parameter1,parameter2)`

When using the second type of command, replace the parameter definitions (between brackets) with their values.

A few rules:

- A single command, comment or label to a line.

- The script analyzer isn't "case sensitive", i.e. upper- and lower-case letters are the same to it.

- Since comma is used to separate parameters, if you whish to insert a comma into a text, first insert a backslash before the comma ("\,"). This way, it won't be interpreted as a separator.

Reminder about coordinates:

X1 and Y1, X2 andY2 are the coordinates of the two opposite corners of a rectangular area (e.g. a screen, a picture, a zone on a tablet, etc.)

LABEL	DESCRIPTION
x1y1 x2y2	X1 and Y1 coordinates correspond to the **upper left-hand corner** of the rectangle. X2 and Y2 coordinates correspond to the **lower right-hand corner** of the rectangle.
X1	Coordinate of the left edge of the rectangle.
Y1	Coordinate of the upper edge of the rectangle.
X2	Coordinate of the right edge of the rectangle
Y2	Coordinate of the bottom edge of the rectangle.

The script commands are presented and explained below, divided into different function categories.

Files and directories

Can a participant's existing data file be overwritten?

`SetSafeRec(DontErase)`

This command redefines the "*No recording overwrite"* option in the dialog box.

Replace "*DontErase*" by:
- TRUE: no recording allowed if a file already exists.
- FALSE: allow data overwrite if one already exists.

<u>Example</u>

If you insert the command `SetSafeRec(False)` into a script, after this command, there will be no further overwrite checks.
If, a few lines further, you insert the command `SetSafeRec(True)`, **from that line onwards**, data will not be erased and it will be impossible to record them with the same name.

Hint: deactivate this checking for the training trials (to limit the number of useless files) and reactivate it for the following experimental trials.

Opens tablet and/or eye-tracking data recording file

`OpenRec(AddToParticipantName)`

This command opens one/two file(s) to record the participant's data: one for the tablet data and one for the eye tracker data, if an eye tracker is used.

The *"AddToParticipantName"* parameter allows you to define a suffix that will be added to the data filename (defined in the acquisition start dialog box).

> **<u>Caution</u>**: each file opened with "*OpenRec*" must be closed with "*CloseRec*" (see next command description). If not, you may lose data.

Hint: to avoid overwriting files if you are using a loop (e.g. when the participant has to perform the same task repeatedly), add "*_%I%"* at the end of *"AddToParticipantName"*. This special option will be replaced by the number of times the script has read the last label (for a definition, p. 66).

<u>Example 1</u>

For a participant called "Toto" (you have defined his name in the start dialog box), the command `OpenRec(_series1)` will record tablet data in the file named "*toto_series1.tab"*.

<u>Example 2</u>

You have used the command `OpenRec(_series1_%I%)`. Participant Toto has passed the same label in the script twice. You will therefore end up with two recording files: "*toto_series1_1.tab*" and "*toto_series1_2.tab*".

CloseRec

This command closes the current participant's recording data file(s) (tablet ".TAB" file and optional eye tracker ".EYE")

SetPicsDirectory(DirectoryPath)

This command redefines the directory of stimuli used in the scripts, initially defined in the "Script" tab of the acquisition configuration panel.

Replace "*DirectoryPath*" with the path to the directory containing your stimuli.

Example
  SetPicsDirectory(c:\mypictures\) gives the "mypictures" directory on the "C" drive of the PC as the new default stimuli directory.

SetDataDirectory(DirectoryPath)

This command redefines the directory in which data files will be recorded (initially defined in the "Script" tab of the acquisition configuration panel).

Replace "*DirectoryPath*" with the path to the directory you want to use.

Example
  The command SetDataDirectory(c:\mydata\) gives the "mydata" directory on the "C" drive of the PC as the new data recording saving directory.

Defines a label
:label

A label identifies a particular place in the script to which you will later refer or for which you will establish a link. Strictly speaking, this is not a command, but a mark in the script. For this reason, each label in the script must be unique.
The "label" notion can be likened to the notion of "bookmark" used in a number of common applications (word processor, etc.) or even with the "anchor" notion used in HTML (Web) documents.
Different "jump" commands (see below) allow the script to discontinue its sequential execution (one line after the other) and go straight to a particular label and then continue executing commands from this location (see other commands in this section).
When "jumping", commands located between the jump command and the destination label are ignored.

Replace the term "*label*" with a word of your choice.
You may define as many labels as you like.

Notion of "counter"

Each label has an internal counter.
The first time a label is "read" by the script interpreter, it sets the label counter to 1. Each time the interpreter reads this label again (like a command, i.e. the label is on a line of its own) for example, because of a jump command, its counter value is increased by 1 (it will then be set at 2, and the next time at 3, etc.).
A label counter's value represents the number of iterations (the action of repeating same processes).

Operations

Some specific commands (see below) allow you to explicitly modify a counter's value. Please remember that performing operations on counter values may lead to negative values, which are not suitable for indexing a list, for example.

Syntax

You can insert spaces in front of the labels or commands (indentation) in order to make your script easier to read.

Example
```
Command1
Command2
    :label
     Command3
     Command4
  etc.
```

Hints:

1. a label can be used as a convenient way of memorizing a value while executing a script. In classic programming languages, this is done with a "variable" when the value is likely to change, and with a "constant" when the value will not be modified.

2. to check a label counter's value, you can use the DisplayMsg command. For example, to display the value of the `"Label1"` label counter, you can use
`DisplayMsg(Label1 value is: %I:Label1%,2000,-1,-1,TRUE)`
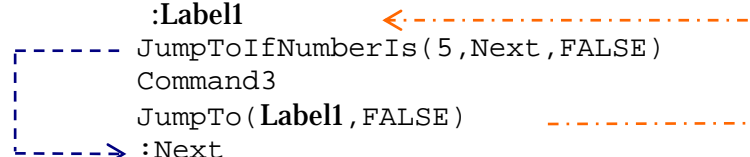
Resets a label counter to zero
ResetLabelCounter(Label)

This command allows you to initialize or re-initialize a label's iteration counter. Replace the word "*Label*" with the name of the label you wish to reset (minus ":").

Example

```
                    :Loop
                     ResetLabelCounter(label1)
                     :Label1
        ------ JumpToIfNumberIs(5,Next,FALSE)
        |          Command3
        |      JumpTo(Label1,FALSE)
        -----> :Next
                     Command4
```

When the script reaches the command `JumpTo(Label1, FALSE)`, it will jump to the "`:Label1`" label.

At the fifth reading of the command `JumpToIfNumberIs(5,Suite,FALSE)` this command will be "activated". The script will then execute it and jump to the "`:Next`" label. `Command3` and `JumpTo(Label1, FALSE)` will not be executed. The script will continue from the "`:Next`" label, and read and execute `Command4`.

In case `Command4` includes a statement to jump to the "`:Loop`" label, Label1's counter should be reset to allow the script to execute the commands following `JumpToIfNumberIs` a further 4 times.

For example, if we imagine a setup with two zones on the tablet, if the participant presses the pen in the first zone the script continues, whereas if the participant activates the second zone, the script jumps to "Loop".

Example of use:

Allows a participant to start a task over again.

Sets the value of a label's counter
SetLabelCounter(Label,Value)

This command allows you to define the value of a label's internal counter.

Replace "*Label*" with the name of the label (minus ":") whose counter value is to be modified.

Replace "*Value*" with the number you wish to assign to this label's counter. "*Value*" may also be replaced with another label's counter value (see "Keywords", p. 87).

Example

`SetLabelCounter(Label1,4)` assigns the value 4 to "*Label1*"

Note:
`SetLabelCounter(Label1,0)` is equivalent to
`ResetLabelCounter(Label1)`

Examples of use

❖ When used in conjunction with `JumpToIfNumberIs`, it allows you to change the script blocks that are to be executed, depending on what the participant

does;

❖ Allows you to start a task again, but with a different number of trials;

❖ Allows you to define a loop using only part of a list, for example only using items 8 to 15 (see commands and keywords for handling a list of items).

Adds to the value of a label's counter

AddToLabelCounter(Label,Value)

This command allows you to add a value to the value of a label's internal counter.

Replace "*Label*" with the name of the label (minus ":") whose counter value is to be modified.

Replace "*Value*" with the number you wish to add to this label's counter. "*Value*" can also be replaced with another label's counter value (see "Keywords", p. 87).

<u>Example</u>
```
AddToLabelCounter(Label1,4)
```

Subtracts from the value of a label's counter

SubtractFromLabelCounter(Label,Value)

This command allows you to subtract a value from the value of a label's internal counter.

Replace "*Label*" with the name of the label (minus ":") whose counter value is to be modified.

Replace "*Value*" with the number you wish to subtract from this label's counter. "*Value*" can also be replaced with another label's counter value (see "Keywords", p. 87).

<u>Example</u>
```
SubtractFromLabelCounter(Label1,4)
```

Display

> **Warning**
>
> **Resolutions and sizes**
>
> When using a masking technique, if you intend to use pictures as a background/wallpaper (e.g. using the *DisplayPic* command), we strongly advise that you adjust the screen resolution to match the picture size. For example, if you are using pictures measuring 1024*768, switch the screen resolution to these dimensions (menu ?/System Information, click on "Screen") and display your pictures at coordinates (0,0).
> It should be borne in mind that the pen's position on the tablet is mapped onto the display (screen) size.
> A different frame and coordinate system for the screen may make some of the participant's writing look distorted.
> Thus, a screen configured with a 1280*1024 pixel resolution, using a display/writing zone (*Configuration/Script* menu) of 1024*768 pixels and displaying a centered picture of 800*600 pixels will produce a "shrinking" effect. Obviously, if this what you are aiming for, forget everything that has just been said!
>
> **Masking/unmasking**
>
> Masking/unmasking production can take two forms: either the "unmasked area" follows the pen location or it remains at (a) fixed location(s) on the screen that is unmasked when the pen reaches that point. Only one technique can be used at a time. If you define unmasking zones (areas), this mode will always take precedence over the "floating" version.

Redefines the coordinates of the display area on the screen
`SetDisplayWindowCoord(X1,Y1,X2,Y2)`

This command redefines the coordinates of the display area on the screen (initially defined in the "Script" tab of the acquisition configuration panel).

Replace *X1, Y1, X2 and Y2* with the coordinates (in pixels) of the new display area you want to use (to define coordinates, see p. 42).

Example
> `SetDisplayWindowCoord(0,0,1024,768)` defines a display area on the screen of 1024 x 768 pixels, beginning in the upper left-hand corner.

Hint: If you have two 1024 x 768 resolution screens, the command `SetdisplayWindowCoord(1024,0,2048,768)` will move the display area to the second monitor. For this to work properly, ensure that Windows desktop extends to both monitors. However, note that this configuration can lead to problems with certain LCD tablet drivers (see Appendix II).

Redefines the screen's background color
`SetBackGroundColor(ColorValue)`

This command redefines the screen's background color (initially defined in the "*Script*" tab of the acquisition configuration panel).

Replace `ColorValue` with the number of the color you want.
Color numbers follow the Web standard and range from #**000000** (**black**) to #**FFFFFF** (**white**).

As a result of the command `SetBackGroundColor(#000000)`, the screen background color will be black.

Redefines the color of the trace left by the pen on the screen

SetPenColor(ColorValue)

This command redefines the "ink" color of the pen when tracing is reproduced on the screen (initially defined in the "Script" tab of the acquisition configuration panel).

Color numbers follow the web standard and range from #**000000** (**black**) to #**FFFFFF** (**white**).

Example: after the command `SetPenColor(#000000)`, the pen will write in black.

Redefines the characteristics of messages or texts displayed on the screen

SetFont(FontName,FontSize,FontColor,BkgndColor)

This command redefines the characteristics of the font and the background of the text displayed on screen (initially defined in the "Script" tab of the acquisition configuration panel).
Replace the parameters (described below) with the appropriate values.

LABEL	DESCRIPTION
FontName	Name of the font used. Respect upper/lowercase letters and spaces in the names.
FontSize	Font size (in points, same unit as in word processors).
FontColor	Font color (color number)
BkgndColor	Background color for text display (color number)

Example

After the command `SetFont(Comic sans MS,14,#000000,#FFFFFF)` has been read by the Eye and Pen script interpreter, the text will be displayed:

- in black (*BkgndColor* = "#000000");
- in the Comic sans MS font (*FontName* = "Comic sans MS");
- in 14 points (*Fontsize* = "14");
- against a white background (*BkgndColor* = "#FFFFFF"*).*

Designates the file to unmask the display

SetUnmaskFile(MaskFileName)

This command allows you to select a picture file that will be used to unmask an area around the pen location. In other words, this is a "peephole" file that helps writers to "see through" the mask covering the screen display.

An "unmasking" file is a two-color (black and white) BMP format picture, where the white area represents the transparent area. Several samples are provided in the "Cursors" folder (the number included in the filename is the "whole" size, expressed in screen pixels). The file is always centered on the pen tip's location.

Replace `MaskFileName` with the filename you wish to use.

Example

The following command identifies the file named "`Oval_asym_right170.bmp`" as the unmasking file:

`SetUnmaskFile(..\cursors\Oval_asym_right170.bmp)`

Note:

The picture file is located not in the *Stimuli* folder, but in the *Cursors* folder. This means that a relative path starting from the *Stimuli* folder has to be used (for more information about relative paths, see p. 15).

Defines how the screen will be masked

SetMaskingMode(Mode)

This command allows you to determine which kind of mask is to be applied to the screen display, out of a possible 3:

- 1 (COLOR): the display is filled with a uniform color
- 2 (PICTURE): the display is concealed by a picture
- 3 (BLUR): the display content is blurred.

Replace "`Mode`" with the mode number you have selected.

Example

The following command sets the screen masking mode to "blur":

`SetMaskingMode(3)`

Note:

If you write your script with the help of the script editor, you will be able to select the mode you require from a scrolling list. Its number will automatically be inserted into the resulting command.

Also see the warning about masking, p. 69.

Selects the picture that will mask the display

SetMaskingFillPic(PictureFileName)

This command allows you to select the picture file that will cover the screen display (see p.13 for a list of supported picture file formats).

Replace `PicturFileName` with the picture filename you want.

Example: `SetMaskingFillPic(MaskFile.jpg)`

Also see the warning about masking, p. 69.

Selects a color to mask the display content

SetMaskingFillColor(ColorNumber)

This command allows you to select the color that will fill the screen display.

Replace `ColorNumber` with the number of the color you have selected.

Note: if you are using the script editor, you will select the color from a color map.

Example
The following command determines the color that is to fill the screen (i.e. orange):
```
SetMaskingFillColor(#FF8000)
```

Sets blur intensity to mask the display
SetMaskingBlurLevel(Intensity)

This command allows you to adjust the intensity of the blurring that will be applied to the screen content.

Replace Intensity with the intensity level you have selected (from 1 to 100).

Because the blurring of the trace left by the pen is dynamically calculated, we advise you to choose a value between 3 and 25. Values above 25 considerably increase the calculation time and hence the time needed to refresh the display.

Example
The following command sets the blurring level at 12:
```
SetMaskingBlurLevel(12)
```

Activates/deactivates display masking
ActivateFeedbackMasking(Active)

This command allows you to turn the screen display masking feature on or off for the following commands: WaitForTabZoneAt, WaitForTabZones and RecStandard.

Replace Active with TRUE to activate the display masking feature and with FALSE to switch it off.

Example: ActivateFeedbackMasking(TRUE)

Defines an unmasking zone
DefineUnmaskZone(x,y)

This command allows you to define an unmasking zone, i.e. a fixed location on the screen that will be unmasked when the pen reaches it.

X and Y represent the upper left-hand corner of the area that is to be unmasked (in pixels). The horizontal and vertical sizes of the area are implicit; they are those of the unmasking picture.

Zones have an order: they are processed in their order of creation.

Important
If unmasking zones are created, they take over the "free" unmasking mechanism. Thus, the unmasking will not freely follow the pen position but will only be effective in some areas of the screen, i.e. the unmasking zones.

Example
The following command creates an unmasking zone at screen coordinates x=100 and y=236 : DefineUnmaskZone(100,236)

Example of use
Create a series of unmasking zones, each one corresponding to a place where the participant is expected to write words. A zone only becomes visible (unmasked) for for the time it takes for a word to be written in it. As soon as the pen leaves the zone,

it is masked again.
The following commands allow you to manage and fine tune the zones' behavior.

ClearUnmaskZones

This command deletes all the unmasking zones that were previously defined.

SetUnmaskZonesRange(LowLimit,HighLimit)

This command defines the boundaries of the zones that are activated (unmasked) at the same time as the zone under the pen tip.

Replace `LowLimit` with the preceding zone's number (in zone creation order, see *DefineUnmaskZone* command) and `HighLimit` with the number of the next zone, relative to the active zone.
The active zone (under the pen tip) is always numbered 0.
The zone preceding the active zone (in order of creation) is numbered -1, the one before that is numbered -2, and so on.
Following the same principle, the zone after the active zone is numbered 1.
There is no limit to the number of previous and subsequent zones that can be activated (unmasked) at the same time.

Example
The following command means that each time a zone is unmasked (with the pen), the previous zone (in order of creation) is also unmasked:
SetUnmaskZonesRange(-1,0)

Note: the default range is (0,0): only the activated-by-pen zone is unmasked.

SetUnmaskZonesBackTracking(Allowed)

This command determines whether or not it is possible to go back, i.e. to unmask a zone that has previously been unmasked (in zone creation order).

Replace `Allowed` with TRUE to enable you to go back or with FALSE to disallow it.

Example
In the following example, if the participant unmasks first Zone 1, then Zone 2, he or she cannot then go back to Zone 1 and unmask it again:
SetUnmaskZonesBackTracking(FALSE)

Note: the default value is TRUE: it is possible to go back.

MustWriteToActivateUnmaskZone(MustWrite)

This command allows you to determine whether the participant has to press the pen tip in a zone to unmask it.

This command is specific to zone unmasking.

Replace `MustWrite` with TRUE to make the pen pressure mandatory, or with FALSE so that the pen's presence above a zone is enough to unmask it.

Note: this value is TRUE by default: the pen must be pressed in a zone for the latter to be unmasked.

Example: `MustWriteToActivateUnmaskZone(TRUE)`

Saves the screen content in an image file
`SaveScreenToBMP(PictureFileName)`

This command converts the screen into a picture file (BMP format).

Replace `PictureFileName` with the name of the picture.

Note: the picture is saved in the *Stimuli* folder, thus allowing you to display it again through a script command.

<u>Examples</u>

1. the command `SaveScreenToBmp(Copy1)` captures the screen content in the file named "*Copy1.bmp*"
2. *for a participant named "Toto",* the command `SaveScreenToBmp(%S%_Capture)` will save the screen in a file named "*toto_Capture.bmp*"

Warning

The script commands presented below allow you to manipulate the way in which the pen's trace is displayed on the screen. Bear in mind that using more than one of these "special effects" at the same time may produce some surprising results. Please also remember that the behavior of the masking/unmasking commands (see previous section) is linked to the pen tip's position on the screen.

Sets the delay for the pen trace's display on the screen
SetTabTimeShiftDelay(Duration)

This command determines the time it takes for the pen's trace to appear on the screen.

Replace `Duration` with the delay value (in milliseconds) you require.

Example

The following command delays the pen trace display on the screen by 300 milliseconds: `SetTabTimeShiftDelay(300)`

Activates/deactivates the delayed display of the pen's trace on the screen
ActivateTabTimeShift(Active)

This command allows you to activate or deactivate the delay in displaying the pen's trace on the screen. This function is associated with the following commands: `WaitForTabZoneAt`, `WaitForTabZones` and `RecStandard`.

Replace `Active` with TRUE to activate the delay and with FALSE to deactivate.

Example

The following command activates the delayed display of the pen's trace on the screen: `ActivateTabTimeShift(TRUE)`

Shifts writing display on screen
SetTabOffset(x,y)

This command allows you to change the reference location for the display of the pen's trace on the screen. This function is associated with the following commands: `WaitForTabZoneAt`, `WaitForTabZones` and `RecStandard`.

Replace `X` with the horizontal offset value (in pixels) you want. A positive value shifts to the right, whereas a negative value shifts to the left.

Replace `Y` with the vertical offset (in pixels) you require. A positive value shifts toward the bottom, whereas a negative value shifts toward the top.

Example

The following command moves the display of writing up by 50 pixels, relative to the location where it would normally be displayed:
`SetTabOffset (0,-50)`

To revert to a "normal" display: `SetTabOffset(0,0)`.

SetTabRatio(Xratio,Yratio)

This command allows you to modify the relationship between the pen movements on the tablet and the way they are displayed on the screen. As such, it allows you to enlarge or shrink the pen trace display on the screen.
Xratio and Yratio are expressed as a percentage of the normal ratio. Thus, a value between 1 and 99 will shrink the trace size, a value of 100 will not change anything and a value above 100 will enlarge the writing display.
This function is associated with the following commands: WaitForTabZoneAt, WaitForTabZones and RecStandard.

Replace Xratio with the horizontal modification factor value you want and Yratio with the vertical modification percentage that is needed.

Example
The following command displays the writing trace at 90% of its normal size, both horizontally and vertically:
SetTabRatio(90,90)

To revert to the "normal" display: SetTabRatio(100,100).

Note

Since modifying the display proportions affects the display surface area (virtual) but not the screen size (physical), enlarging the writing gives the impression that the writing has shifted to the bottom right-hand corner of the screen, whereas shrinking the writing will "shift" it to the top left-hand corner. For example, if the size of the writing is doubled with SetTabRatio(200,200), the point that should have been displayed at screen coordinates (20,20) will be displayed at coordinates (40,40). Think of it as though the size of the display had been doubled but only the upper left-hand quarter was still visible on the screen, because the screen's size cannot be doubled as well.
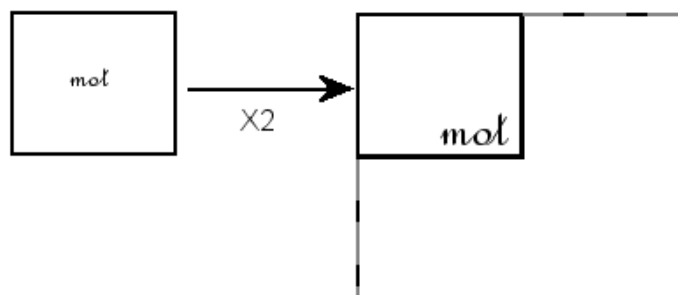
Redefines the calibration area coordinates on the screen
SetCalibrationCoord(X1,Y1,X2,Y2)

This command redefines the calibration area coordinates on the screen (initially defined in the "Eye tracker" tab of the acquisition configuration panel).

Replace *"X1", "Y1", "X2" and "Y"* with the calibration area coordinates (in pixels) you have chosen (see p. 63).

<u>Example</u>

`SetCalibrationCoord(0,0,1024,768)` defines a calibration area that is 1024 pixels wide (horizontally) and 768 pixels high (vertically), beginning in the screen's upper left-hand corner.

Hint: If you are using two 1024*768 resolution screens, the `SetCalibrationCoord(1024,0,2048,768)` command will locate the calibration area on the second (screen) monitor.

Launches the calibration procedure
TestCalibration

This command launches the calibration procedure for the selected eye tracker;
For Eyelink, the calibration procedure is supplied by the eye tracker interface.
For Eyeputer, this command launches a specific ("home made") calibration dialog box (see p. 184).

Tests calibration reliability
TestDrift

This command launches a test to assess eye tracker coordinate drift.

For Eyelink, the test is run with a single (central) point, whereas with EyePuter and ASL504, the entire calibration procedure is repeated. For EyePuter, you can also choose to test just one point.

Pauses the script for a certain duration
WaitFor(duration)

This command stops the script execution for « duration » milliseconds.
When the delay is elapsed, the script execution goes on again.

Example
The command WaitFor(2000) stops the script for 2000 milliseconds, i.e. 2 seconds. When this delay ends, the script continues.

Stops the script. It starts again when a keyboard key is pressed
WaitForKeyPress

This command puts the script on hold. As soon as a key is pressed, the script execution resumes.

Displays a message on the screen and stops the script until a keyboard key is pressed
WaitForKeyPressMsg(Message,X,Y,Transparent)

This command displays a message on the screen and stops the script. As soon as a key is pressed, the message disappears and the script resumes.

To set the message content, replace the term "*Message*" with the message content you want.

Caution: do not insert "**,**" in your message, as it is a parameter separator and the command may have unexpected consequences.

To set the message's position in the display windows, replace "*X*" with the horizontal coordinates (in pixels) of the message's first character and "*Y*" with its vertical coordinates.

To set the message's horizontal and vertical coordinates, you can:
- proceed by trial and error;
- use a relation including the screen's physical size and its resolution (see FAQ, p. 199).

Hint: to center the message in one/both dimension (s), set the corresponding coordinate to "-1*".*

Replace **Transparent** with FALSE if you want the message background to be filled with the current font background color, or with TRUE to have it display in transparency.

Example
The command WaitForKeyPressMsg(Press a key to continue..., -1,600,FALSE) stops the script and displays the message "Press a key to continue..." on the screen. This message is horizontally centered (X=-1), near the bottom of the screen (its vertical coordinate Y is 600). As soon as a key is pressed, the script resumes.

`WaitForKeyPressText(TextFileName)`

This command displays the content of a text file on the screen (taken from the stimuli directory) and stops the script. As soon as a key is pressed, the text disappears from the screen and the script resumes.

<u>Recall</u>: the stimuli directory is defined in the "*Script*" tab of the acquisition configuration panel, or redefined via the `SetPicsDirectory` command.

To be properly displayed, the text file must be in a "text file" format (".TXT" extension), or "text only" or "raw text", depending on the application.

<u>Example</u>

The command `WaitForKeyPressText(mytext.txt)` displays the text contained in the "mytext.txt" file and stops the script.

`WaitForKeyPressPic(PictureFileName,X,Y)`

This command displays the picture (found in the stimuli directory) and stops the script. A s soon as a key is pressed, the script resumes.

To set the picture that will be displayed, replace "*PictureFileName*" with the name of the file containing the picture. The picture must be in a bitmap format ".bmp".

To set the picture position in the display window (on the screen), replace "*X*" with the horizontal coordinates (in pixels) of the picture's upper left-hand corner, and "*Y*" with the vertical coordinates of the picture's upper left-hand corner.

To set the picture's horizontal and vertical coordinates, you can :
- proceed by trial and error;
- use a relation including the screen's physical size and resolution (see FAQ, p. 199).

Hint : to center the picture in one/both dimension(s), set the corresponding coordinate to "-1*".*

<u>Example</u>

The command `WaitForKeyPressPic(icon.bmp,10,370)` displays the "icon.bmp" picture at the following coordinates: 10 (horizontal) and 370 (vertical).

`WaitForTabZoneAt(x1,y1,x2,y2,CanDraw,MustLeave)`

This command stops the script until the participant presses the pen in the tablet area defined in this command. If the parameter CanDraw is activated, the paticipant's writing is displayed on screen. This command is thus a mean to have a participant write (draw) on the tablet whilst displaying his or her writing's trace on screen until he or she presses the pen in a defined area (zone).

To define the tablet zone (area), replace *"X1", "Y1", "X2" and "Y2"* with the coordinates (in tablet units, i.e. lines) of the area you have chosen (see p. 14).

- With the 4 coordinates set to **0**, the pen may be pressed anywhere on the tablet to "re-start" the script.
- With the 4 coordinates set to **-1**, it is the **start zone** coordinates which are used (defined in the "Script" tab of the acquisition configuration panel)

The other parameters can be given two values :
- "TRUE": activates the parameter;
- "FALSE": deactivates the parameter.

The table below explains these parameters, assuming a "TRUE" value.

LABEL	DESCRIPTION
CanDraw	The participant's writing is reproduced on the screen, until the pen is pressed in the zone.
MustLeave	The pen must leave the zone before the script can continue.

Example

```
WaitForTabZoneAt(17327,5015,19850,2415,FALSE,TRUE,FALSE)
```
stops the script. If the participant presses the pen in the defined tablet area (coordinates X1=*17327*, Y1=*5015*, X2=*19850*, Y2=*2415),* the script resumes.

The NetSync module allows you to drive a recording session on several computers via network commands (see Netsync, p. 104).
The commands shown below should only be used in this context.

Stops the script until a signal is sent by the Master host
WaitForNetSync

This command halts script execution.
Script execution resumes when the "Go" signal is received from the Master host.

Sends a message to the Master host
SendMessageToNetSync(Message)

This command allows you to send a message to the Master host. This message is displayed in the "State" column of the NetSync data grid holding the client hosts.

Examples of use
-   Show (on the Master host) the item currently being processed on the client host;
-   Follow the stages of the experiment on each client host;

Example
The command `SendMessageToNetSync(item 3)` sends the message "item 3" to the Master host.

The message may include or be based on one or more keywords (see *Keywords* section, p. 87). For example, `SendMessageToNetSync(%L%)` sends the current list item; `SendMessageToNetSync(item %I%/%M%)` sends the label counter value and the number of items in the list (e.g. "item 5/12").

The mechanism of **jumping into/through the script** allows you to eliminate the sequential aspect of the script (one command after another, in the right order) and create loops or interactive sequences.
This enables a participant to repeat the same task many times (with a maximum number) or to carry out different tasks, depending on the zone in which the pen is pressed.

Jumps to a label in the script

JumpTo(Label,MustCloseRec)

This command "jumps" to the relevant label and, if specified in the parameters, closes the recording file.
All commands between this command and the relevant label will be ignored and the script will resume after the label.
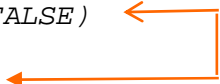
Replace "*label*" with the name of the label you want the script to jump to (minus ":").
Replace "*MustCloseRec*" with "TRUE", if you want to close an open data recording file, or "FALSE" if you do not need to.

<u>Example</u>

```
        Command2
 JumpTo(Zone1Go,FALSE)
        Command3
       :Zone1go
       command5
```

When the script interpreter reads the command `JumpTo(Zone1Go,FALSE)`, it will jump to "*Zone1go*". Commands between `JumpTo(Zone1Go,FALSE)` and the destination label will not be executed (`command3`). The script will continue and execute `command4`. If the "MustCloseRec" parameter had been given the "TRUE" value, the participant's recording file would have been closed.

This command is mostly useful when used in conjunction with conditional jump commands (a jump is made if a condition is met, e.g. when the pen is pressed in a particular area of the tablet).

Jumps to a "target" label when the script has passed the last label (preceding the command) a certain number of times

JumpToIfNumberIs(Iterations,Label,MustCloseRec)

This command is only executed if the script has passed the label in front of (above) the command a certain number of times. If this is the case, the jump will be executed to the relevant label.

Replace "*Iterations*" with the maximum number of times the script can "see" the preceding label before executing the jump.

Replace "*Label*" with the name of the label to jump to (minus ":").
Replace "*MustCloseRec*" (close an open recording data file or leave it open) with "TRUE" to activate this parameter or "FALSE" to ignore it.

```
:BlocBegins
Command2
JumpToIfNumberIs(5,Next,FALSE)
 Command3
JumpTo(BlocBegins,FALSE)
:Next
Command4
```

When the script reaches the `JumpTo(BlocBegins,FALSE)` command, it jumps to the ":`BlocBegins`" label. The `JumpToIfNumberIs(5,Next,FALSE)` command is activated after its fifth "reading". The command is then executed and the script jumps to the ":Next" label. Command3 and `JumpTo(BeginningOfBloc, FALSE)` are ignored and not executed. The script resumes from the ":`Next`" label, reads command4 and executes it.

Examples of use

1 – Making a participant repeat a task several times.

2 – Used in conjunction with other commands, allowing a participant to repeat a task a maximum number of times, without any obligation to reach this maximum. By the end, `Command2` will have been executed 5 times, whereas `Command3` will have been executed 4 times.

Jumps to a "target" label when a given label's counter reaches a given value

JumpToIfLabelIs(DestinationLabel,LabelToCheck,Iterations, MustCloseRec)

This command is only executed if the designated label "to watch" has been read a certain number of times (if its counter reaches a predetermined value).
The script execution then "jumps" to the "target" label's location in the script.

Replace `DestinationLabel` with the name of the label to jump to (minus ":").

Replace `LabelToCheck` with the name of the label (minus ":") whose counter is to be checked against a predetermined value.

Replace `Iterations` with the value to "look for".

Replace `MustCloseRec` (close an open recording data file or leave it open) with **"TRUE"** to activate this parameter or "FALSE" to ignore it.

Note
   This command is similar to the `JumpToIfNumberIs` command, except that instead of implicitly using the current label's (last "seen" label's) counter value, a label name is explicitly given to serve as a criterion. It is possible to monitor the value of any label defined anywhere in a script.

LoopIfLabelIsBelow(Label,Iterations,MustCloseRec)

This command is executed if the "watched" label's counter value is below the value specified in the command parameters.
This command allows you to jump to the "watched" label.

Replace `Label` with the name of the label to "watch" (minus "**:**") and to jump to when needed.
Replace `Iterations` with the threshold value.
Replace `MustCloseRec` (close an open recording data file or leave it open) with **"TRUE"** to activate this parameter or "FALSE" to ignore it.

Use
The basic function of this command is to allow you to perform loops, i.e. a block of actions you want to have performed a certain number of times.

Example

```
----->  :BlockStart
           Command1
           Command2
           Command3
------     LoopIfLabelIsBelow(BlockStart,5,FALSE)
         Command4
```

When the script interpreter reads the command
`LoopIfLabelIsBelow(BlockStart,5,FALSE)`, it checks the value of the internal counter of the "BlockStart" label. If this value is below 5, it jumps to the "BlockStart" label; otherwise it goes on to execute the next command `Command4`.

Example of use

Make a participant perform the same task more than once.

Notes
- When the script interpreter "reads" the "`BlockStart`" label for the first time, its internal counter is set at 1.
- You can modify a label's counter value with the help of the `SetLabelCounter` command.

Using a list of items may help either to simplify the process of writing a script or, conversely, to develop a sophisticated experiment.
An Eye and Pen script automatically has a single list.
Each item in the list may be a word, a sentence, a number, a filename, etc. In short, anything that can be written using letters and numbers.

Adds an item to the list
AddToList(ItemName)

This command allows you to add an item to the list. This new item is appended to the end of the list. The list is not sorted in any specific way: items appear in the order in which they are added.

Replace `ItemName` with the item that is to be added to the list: a string of letters and/or numbers (word, sentence, 500, etc.)

Examples
```
AddToList(This is a sentence to read)
AddToList(1200)
AddToList(image3.wmf)
AddToList(sound.wav)
```

Loads an item list from a file
LoadList(FileName)

This command allows you to load a set of items contained in a text file. Its content is then included in the list.
Items are appended to the end of the list. Loading from a file does not delete items previously included in the list.

Replace `FileName` with the name of the file to be loaded.

The file must be in a ".TXT" text-only format and should contain one item per line. The file must be located either in the *Stimuli* folder or else at the end of a relative path leading from this folder (cf. p. 15).

Example:
```
LoadList(liste.txt)
```

Shuffles item order in the list
RandomizeList

This command allows you to shuffle the items around in the list.

Shuffles item order in part of the list
RandomizeListRange(Begin,End)

This command allows you to shuffle items in a random order in just part of the list.

Replace `Begin` with the number of the first item in the part of the list that is to be shuffled, i.e. at the beginning of the set to be randomized.

Replace `End` with the number of the last item in the set to be randomized.

Note: position numbering in the list starts with 1.

<u>Example</u>

The command `RandomizeListRange(3,9)`randomizes all items from positions 3 to 9 (inclusive).

<u>*Example of use*</u>

To be able to use a list of items within a test session divided in sub-blocks, the items in each one being randomized. For example, while the list's first items (practice) are not randomized, two successive blocks of items are randomized, all in a single list. This makes it easier to write a script because the experimental session can simply run from item 1 to the end of the list.

Clears the list
ResetList

This command empties the list of its contents: all items are deleted.

Keywords are special words that are replaced by a "definite" value during script execution. Keywords can be likened to the notion of *variable* that is used in common programming languages, except that a keyword cannot be modified by hand, being "read-only" (its value can only be "read" by a script).
Keywords can be included in every parameter of every script command.

Takes the value of the current label's internal counter

`%I%`

> This keyword is replaced with the value of the current label's counter (the last label read in the script).

> Example
>> The following command adds the value of the current label's counter to the name of the protocol recording filename (participant id.): `OpenRec(_%I%)`

Retrieves an item from the list in the position corresponding to the current label's counter value

`%L%`

> During script execution, this keyword is replaced with the content of an item retrieved from the list (the item is the one whose position in the list corresponds to the current label's counter value).
> `%L%` may be included in the parameters of all the commands.

> Example
>> The following script adds items to the list, randomizes the list, and has the items displayed on the screen. This example shows how to create a loop to manage the list content. Thanks to `%L%`, the script will remain the same for 100 items.

```
ResetList
```

; add two items: picture filenames
```
AddToList(image1.bmp)
AddToList(image2.bmp)
```

; shuffle the entire list
```
RandomizeList
:Start
```

; %L% retrieves the item at position "n" in the list, "n" being the "Start" label'
; counter value.

; The command displays this picture (its name has just been retrieved from the list)
; for 2000 milliseconds, centered on the screen.
```
 displaypic(%L%,-1,-1,2000)
```

; if the counter value of the current label ("`Start`") is 2, then jump to the label
; named "`End`"
```
 JumpToIfNumberIs(End,2,True)
```

; if not, jumpt to "`Start`" for another round
```
JumpTo(Start)
```

```
:End
```

Takes on the number of items in the list
`%M%`

When executing the script, this keyword is replaced with the number of items found in the list (list size).

Example
The following command may be used to exit a loop when all the items have been processed:
`JumpToIfNumberIs(End,%M%,TRUE)`

Takes a random number
`%R`*`Value`*`%`

When executing the script, this keyword is replaced with a random number. Replace `Value` with the upper limit of the range within which the random number must be taken. The range goes from 0 to `Value`.

Example 1
The following command waits for a random duration of between 0 and 499 milliseconds (inclusive): `WaitFor(%R500%)`

Example 2
Create a delay of 1,000 milliseconds: `WaitFor(1000)`
Create a random delay of between 0 and 500 milliseconds:
`WaitFor(%R501%)`

Participant's name
`%S%`

When executing the script, this keyword is replaced with the participant's ID (provided in the script launching dialog box).

Caution: commands that load a file whose name includes this keyword will not be checked (no such files are supposed to exist at the time of script checking).

Example
The command `DisplayPic(%S%_page1.bmp,-1,-1,-1)` can be used to reload a page previously written by a participant and saved with the command `SaveScreenToBMP(%S%_page1)`.

Uses a label's counter value with a keyword
`%?:Label%`

The label's counter value may be invoked with any keyword (except `%M%`, obviously).

You can therefore "target" a specific label's counter value.

Unlike the syntax used with other commands, here the label should be mentioned with its ":", just as it is shown in the script (alone on a line).

The "**?**" is replaced with a keyword, either **I**, **L** or **R.**

Example 1
The following command uses the value of the "Start" label's counter as an index to retrieve an item in the list: `DisplayPic(%L:Start%.jpg,-1,-1,-1)`

Example 2
The value of the "`Start`" label's counter is used to help "build" the name of a picture to be displayed, e.g. "pic_3.bmp".
`DisplayPic(pic_%I:Debut%.bmp,-1,-1,-1)`

Example 3
The command `WaitFor(%R:label3%)` receives a random value of between 0 and the value of the "`label3`" label counter.

Example 4
The command `WaitFor(%L:label3%)` uses the item in the list whose position corresponds to the value of the "`Label3`" label counter. This then allows you to manipulate delays using the list items.

Example 5
The command `SetPenColor(%L%)` selects the ink color from the list.

Associates a label with a tablet zone

`DefineTabZone(X1,Y1,X2,Y2,Label)`

This command allows you to "bind" the indicated label to a tablet area. When the participant presses the pen into the defined tablet are, the script jumps to the label location (in script).

To define the tablet zone, replace *"X1", "Y1", "X2" and "Y2"* with the area's coordinates on the tablet.

Replace "*label*" with a word of your choice.

Reminder: the coordinates are defined in tablet units (see p. 14).

Example

```
DefineTabZone(7327,5015,1850,2415,Zone1Go)
Command3
Command4
:Zone1Go
command5
```

`DefineTabZone(7327,5015,1850,2415,Zone1Go)` defines a tablet zone with coordinates *X1= 7327*, *Y1= 5015*, *X2=1850* and *Y2=2415*, bound to the "`:Zone1Go`" label.

You can use this command several times to create several zones that will be used at the same time.
This way, you create a "multiple choice" situation.

Zones will be checked in their order of creation. This means that you can create one zone inside another, by creating the smaller zone first, then creating a larger one to encompass it. When the pen is pressed in the smaller zone, it will be recognized as an activation of the smaller zone, not the larger one.

You can create multiple zones aiming towards the same label.

Jumps to a label depending on the selected tablet zone

`WaitForTabZones(CanDraw,MustLeave,MustCloseRec)`

This command stops the script until the participant has pressed the pen in one of the zones previously defined using the `DefineTabZone(X1,Y1,X2,Y2,Label)` command (see previous command description).

The **parameters** can be given two values :
- "TRUE": activates the parameter;
- "FALSE": deactivates the parameter.

The table below explains these parameters, assuming a "TRUE" value.

LABEL	DESCRIPTION
CanDraw	The participant's writing (drawing) is reproduced on the screen until the pen is pressed in one of the zones the script

	is waiting for.
MustLeave	The pen must "quit" the zone before the script can continue.
MustCloseRec	Closes the current data recording file.

Example

```
        Command2
        DefineTabZone(7327,5015,1850,2415,Zone1Go)
        Command3
        WaitForTabZones(TRUE,FALSE,TRUE,FALSE)
        Command4
      :Zone1go
        command5
```

The ":Zone1Go" label *is bound* to a tablet zone by the command
DefineTabZone(7327,5015,1850,2415,Zone1Go)*.*

The script executes Command3 and goes on hold, with the command
WaitForTabZones(TRUE,FALSE,TRUE,FALSE)*.*

As the Candraw parameter is "TRUE", the participant's writing is shown on the screen until the command comes to an end, i.e. when the participant presses the pen in the defined tablet zone.

As the MustLeave parameter is "TRUE", the command will only come to an end when the pen leaves the tablet zone.

The script will then "jump" to the ":zone1Go" label.

The script will continue and execute Command5.

In this example, you can see that command4 will never be executed.

Waits for the pen to leave the active zone
WaitForPenOut(MaxStimulusDisplay,StimulusType)

This command halts script execution until the participant has removed the pen from the zone it is being pressed in. Command parameters allow you to specify a maximum "*display duration*" for each stimulus.

When used jointly with WaitForTabZones, this command allows you to manage the start and end of a stimulus after a given length of time (or earlier, depending on the participant's behavior).

Example
With the command WaitForPenOut(3000,'IMAGE'), each image-type stimulus will be erased from the screen after 3 seconds (3000 milliseconds), or earlier if the participant removes the pen from the zone before the 3 seconds are up.
Script execution will resume as soon as the participant has removed the pen from the zone.

Stimulus type are:

- o IMAGE
- o TEXT (text file)
- o MESSAGE
- o VIDEO

- o AUDIO
- o ALL: every stimulus type (listed above). If several stimuli are simultaneously played (e.g. a picture and a sound), both will be stopped.

<u>Example of use in a script</u>:

; define a tablet zone to trigger the display of a picture
```
DefineTabZone(1,1,120,120,zone_image_1)
```

; define a zone to end the script
```
DefineTabZone(1,200,120,320,zone_end)
```

; jump into the "loop"
```
jumpto(WaitForTabzonesAnchor)
```

; label for "image 1" zone
```
:zone_image_1
```

; display picture image1.bmp for an unlimited duration (-1)
```
DisplayPic(image1.bmp,-1,0,0)
```

; wait for the pen to be removed. After 3 seconds, clear the picture.
```
WaitForPenOut(3000,IMAGE)
```

; label for the "loop"
```
:WaitForTabzonesAnchor
```

; wait for the pen to be pressed in a zone. Parameters: writing on the screen. If the participant selects a zone intended for script, do not wait for the pen to be removed.
```
WaitForTabZones(TRUE,FALSE,FALSE)
```

; label for "end"
```
:zone_end
```

; wait for the pen to be removed. After 2 seconds, stop all stimuli.
```
WaitForPenOut(2000,ALL)
```

With this script, the participant has to write (the pen's trace will appear on screen) and he or she can do so in two zones, each associated with an action:

➢ a zone linked to the label "`zone_image_1`" where the picture `image1.bmp` appears when the participant presses the pen and disappears after 3 seconds, unless the participant removes the pen beforehand (in which case the picture is immediately cleared);

➢ a zone linked to the "`zone_end`" that stops all stimuli after 2 seconds, unless the participant removes the pen beforehand. The script then ends because there are no more commands.

Clears previously defined tablet zones (defined by *"DefineTabZones"*)
ClearZones

This command erases all tablet zones previously defined using the "`DefineTabZone`" command.

Displays a message for a certain length of time
`DisplayMsg(Message,Duration,X,Y,Transparent)`

This command is used to display a message for a user-defined duration, at the indicated screen coordinates. The script remains "on hold" until the display time has elapsed. The message disappears when the command comes to an end.

To set the content of the message, replace "*Message*" with the text you want to display.

Caution: do not include "**,**" in your message, as it is a parameter separator.

To set the message display duration, replace "*Duration*" with the desired amount of time, expressed in **milliseconds.**

To set the message's position in the display windows, replace "*X*" with the horizontal coordinate (in pixels) of the message's first character and "*Y*" with its vertical coordinate.

To set the message's horizontal and vertical coordinates, you can:
- proceed by trial and error;
- use a relation including the screen's physical size and resolution (see FAQ, p. 199).

Hint:
- to center the message in one/both dimension(s), set the corresponding coordinate to "-1";
- to leave the message on the screen "forever", set duration to "-1". The script will resume immediately, without erasing the message.

Replace **Transparent** with FALSE if you want the message to be displayed on a background stripe colored with the current font background color or with TRUE to have only the characters of the message displayed on the screen.

Example
With the command `DisplayMsg(Hello world,500,120,600,TRUE),` the message "Hello world" will be displayed for 500 milliseconds.

Erases the message left on the screen
`HideMessage`

This command removes the message previously displayed with DisplayMsg (see above).

Displays a text file for a certain length of time
`DisplayText(TextFileName,Duration)`

This command is used to display the content of a text file for a certain length of time. The script remains until the display time has elapsed. The text disappears when the command comes to an end.

Replace "*TextFileName*" with the name of the **text file** you placed in the stimuli directory. Do not forget to add the ".txt" extension to the filename.

Replace "*Duration*" with the **duration** (in milliseconds) of the text display.

Hint: if you want the text to remain "forever" on the screen, set the duration to "-1". The script will immediately resume, without erasing the text.

Example

With the `DisplayText(mytext.txt,500)` command, the text contained in the text file named "*mytext.txt*" (found in the stimuli directory) will be displayed for 500 milliseconds.

Erases the text on screen
HideText

This command removes the text displayed on the screen as a result of the "`DisplayText`"command.

Displays a picture for a certain length of time
DisplayPic(PictureFileName,Duration,X,Y)

This command is used to display a picture (found in the stimuli directory) for a certain length of time. The script stays on the screen until the display time has elapsed. The picture disappears when the command comes to an end.

Replace "*PictureFileName*" with the name of the **picture** (do not forget the format extension).

Replace "*Duration*" with **the length of time** (in milliseconds) you want.

To set the picture's position in the display windows, replace "*X*" with the horizontal coordinate (in pixels) and "*Y*" with the vertical ccordinate of the picture's upper left-hand corner.

To set the picture's horizontal and vertical coordinates, you can:
- proceed by trial and error;
- use a relation including the screen's physical size and resolution (see FAQ, p. 199).

Hint:
- to center the picture in one/both dimension(s), set the corresponding coordinate to "-1".
- to make the picture remain on the screen "forever", set the duration to "-1". The script will resume immediately, without removing the picture.

Example

With the `DisplayPic(icon.bmp,500,120,120)` command, the picture "*icon.bmp*" will be displayed for 500 milliseconds at the coordinates *X=120*, *Y=120.*

Erases the picture on the screen
HidePicture

This command removes the picture from the screen (previously displayed using the `DisplayPic` or `DisplayImageList` command). Background is restored.

Displays a video file
DisplayAVI(VideoFileName,X,Y,Wait)

This command displays a video (found in the stimuli directory) at specified coordinates (relative to the display window). The video disappears when it is

finished.

Replace "***VideoFileName***" with the video file's full name (including its ".AVI" extension).

Caution: Eye and Pen can only read ".avi" video format, preferably with MS-RLE compression. To check whether the video file complies:

> ➢ In Windows, select the video file and, with a right mouse button click, select the Summary tab in Properties;
> ➢ Look in "*Video*" for the "*Compression*" label to check the video's compression mode. It should be either "none" or "MS-RLE".

To set the video's position in the display windows, replace "*X*" with the horizontal coordinate (in pixels) of the video's upper left-hand corner and "*Y*" with its vertical coordinate.

To set the video's horizontal and vertical coordinates, you can:

- proceed by trial and error;
- use a relation including the screen's physical size and resolution (see FAQ, p. 199).

Hint: to center the video in one/both dimension(s), set the corresponding coordinate to "-1".

The "Wait" parameter determines whether the script should wait for the video to finish before continuing or not. This parameter can be given two values:

- "FALSE": the video starts to play and the script immediately resumes;
- "*TRUE*": the film starts and the script goes on hold. No script commands are executed while the video is being played.

<u>Example</u>

With the command `DisplayAVI(MyVideo.avi,-1-1,TRUE,)` the "*Myvideo.avi*" video is played in the center of the display window, because "*X*" and "*Y*" both have a "-1" value. The script resumes as soon as the video is finished.

Stops the video playing

StopAVI

This command stops the AVI video currently being played

Displays a series of pictures one after another

**DisplayImageList(ListFileName,X,Y,DurationPerPicture,
                  HideLastPic)**

This command displays pictures one after another (at the specified coordinates) for a specified length of time.

Replace "***ListFileName***" with a text filename (TXT format) containing the picture list (see supported pictures formats, p. 13) to be displayed (one picture name per line). Save this file and the pictures in the stimuli directory.

> **<u>Caution</u>**: all pictures must be the same size. The first picture to be displayed will define the size of all the others.

<u>Example:</u>

The file `MyList.txt` contains:
Eye.bmp
And.bmp

Pen.bmp

To set the picture's position in the display windows, replace "*X*" with the horizontal coordinate (in pixels) of the picture's upper left-hand corner and "*Y*" with its vertical coordinate.

To set the picture's horizontal and vertical coordinates, you can:
- proceed by trial and error;
- use a relation including the screen's physical size and resolution (see FAQ, p. 199).

Hint: to center the picture in one/both dimension(s), set the corresponding coordinate to "-1".

Replace "***DurationPerPicture"*** with the amount of time (in milliseconds) you want each picture to be displayed for.

Replace "***HideLastPic***" with TRUE to remove the last picture from the screen or FALSE to leave it.

Hint: if you use the same background color for all the pictures, you will reduce the visual transition effect between the pictures.

Example

With the *DisplayImageList(Mylist.txt,120,120,250,FALSE)* command, the pictures listed in the "*mylist.txt*" file will be displayed one after the other. Each picture will be displayed at the coordinates X=120 and Y=120 for 250 milliseconds and the final picture will remain on the screen.

Plays a beep
Beep

This command plays the Windows basic sound defined in the Windows Control Panel (under Windows XP "Sounds and audio devices", "Sounds" tab), "default beep" entry.

Makes the system generate a beep
SystemBeep(Frequency,Duration)

This command makes the system generate a beep of the frequency and duration you want.

Replace `Frequency` with the frequency value of your choice (in Hz) and `Duration` with the required duration (in milliseconds).

Caution: this beep is played on through the computer's internal speaker; it uses neither the sound card nor the audio, headset or speaker output of the sound device (card).

Example

The command `SystemBeep(1000,200)` plays a continuous 1,000 Hz frequency sound for 200 milliseconds.

Plays an audio file

`PlaySound(WaveFileName,Wait)`

This command plays an audio file (found in the stimuli directory) in ".wav" format. Replace "*WaveFileName*" with the filename (do not forget to add ".wav" to the name).

The "Wait" parameter determines whether the sound is to be played as a background task or not.

This parameter can be given two values:

- "FALSE": playing begins and the script resumes immediately;
- "*TRUE*": playing begins and the script goes on hold. No other script command is executed until the audio sequence has finished.

Caution:

If you use this command twice (with no other command between the two), a "system" malfunction may occur.

This is because "Eye and Pen" uses a media player which takes some time to close, before being able to reopen with another file.

It is therefore advisable to create a single sound file containing several sound sequences (e.g. as a series of numbers).

Example

The command `PlaySound(ding.wav,FALSE)` plays the "*ding.wav*" sound file and the script continues (the "*Wait*" parameter is "*FALSE*"). The script's next command is immediately executed.

Example of use: a dictation.

Stops the audio play

`StopSound`

This command stops the audio file currently being played.

Changes the sound output level

`SetVolume(value)`

This command allows you to change the sound level for the audio output.

Replace "value" with a number between 0 (mute) and 65535 (maximum) to set the sound level.

The commands described below are intended to give you the ability to use the "Simple" acquisition mode ("Simple" tab of acquisition configuration panel), allowing you to change some or all of its parameters.
You will then be able to record more than one task using this paradigm, for instance, with different pictures each time. Then again, you could use this task without any modification, but in the middle of a set of other tasks.

Launches the Simple acquisition paradigm

`RecStandard(AddToParticipantName)`

The `RecStandard` command allows you to activate the simple acquisition paradigm, with the parameters defined in the acquisition configuration panel (Simple tab) of Eye and Pen. Replace *"AddToParticipantName***"** with a suffix that will be added to the recording file's name.

> **Caution**: this command manages the opening and closing of the recording data files. This means that you do not have to use `OpenRec` and `CloseRec`.

Example
For the participant "Toto", the command `RecStandard(_standard)` launches the Simple acquisition (as if you had launched it yourself, using the File/Acquisition/Simple menu of Eye and Pen). The tablet data will be saved in "*toto_standard.tab"* and the eye-tracking data in "*toto_standard.eye*".

Launches the Simple acquisition, changing background picture and trigger zone use

`RecNewUsages(AddToParticipantName,UseBack,UseZone1,`
`              ShowOnStart,HideOnPress,UseZone2)`

This command allows you to launch the Simple acquisition, redefining the activation of the background picture and trigger zones 1 and 2. These uses were initially defined in the acquisition configuration panel (Simple tab).

> **Caution**: this command manages the opening and closing of the recording data files. This means that you do not have to use `OpenRec` and `CloseRec`.

Replace *"AddToParticipantName"* with a suffix that will be added to the recording file's name.

Parameters can be given two values:
- "TRUE": activates the parameter;
- "FALSE": deactivates the parameter.

The following table describes the parameters, assuming a "TRUE" value.

PARAMETER	DESCRIPTION
UseBack	Use a background picture.
UseZone1	Use trigger zone 1.

ShowOnStart	Show the picture associated with trigger zone 1 as soon as recording starts.
HideOnPress	Hide the picture associated with trigger zone 1 as soon as the participant presses the pen on the tablet. **Caution**: this parameter will only be taken into account if "*ShowOnStart*" is activated.
UseZone2	Use trigger zone 2.

<u>Example</u>

With the participant "*Toto*", the command
`RecNewUsages(_P3,FALSE,TRUE,FALSE,FALSE,FALSE)`:
- saves data under the name "*toto_P3*";
- there will not be any background picture ("*UseBack*" is "*FALSE*");
- trigger zone 1 will be used *("UseZone1"*= "TRUE");
- the associated picture will not be displayed *("ShowOnStart*" is "FALSE" and so "*HideOnPress*" will be ignored);
- trigger zone 2 will not be used *("UseZone2*" is "*FALSE*").

RecNewPics(AddToParticipantName,BackPic,Pic1,Pic2)

This command allows you to launch the Simple acquisition and to change the background picture and pictures associated with trigger zones 1 and 2 (initially defined in the Simple tab of the acquisition configuration panel).

> **Caution**: this command manages the opening and closing of the recording data files. This means that you will not have to use `OpenRec` and `CloseRec`.

Replace *"AddToParticipantName"* with a suffix that will be added to the recording file's name.

The following table describes the parameters.

PARAMETER	DESCRIPTION
BackPic	Background picture filename
Pic1	Picture filename associated with trigger zone 1
Pic2	Picture filename associated with trigger zone 2

<u>Example</u>:

For the participant "*Toto*", the command
`RecNewPics(_P2,YellowBack.bmp,Capsela.bmp,car.bmp)`:
- saves the data under the name "*toto_P2*";
- the background picture is "YellowBack.bmp" ("*BackPic*" is "*YellowBack.bmp*");
- the picture associated with trigger zone 1 is "*Capsela.bmp*";
- the picture associated with trigger zone 2 is "*car.bmp*.

**RecNewPics&Usages(AddToParticipantName,BackPic,UseBack,
Pic1,UseZone1,ShowOnStart,HideOnPress,Pic2,UseZone2)**

This command is a "compilation" of `RecNewPics` and `RecNewUsages` (see above). It launches the Simple acquisition, allowing you to change the pictures and their activation for background and trigger zones at the same time.

> **Caution**: this command manages the opening and closing of the recording data files. This means that you do not have to use `OpenRec` and `CloseRec`.

Replace *"AddToParticipantName"* with a suffix that will be added to the recording file's name.

Parameters for use can be given two values:
- "TRUE": activates the parameter;
- "FALSE": deactivates the parameter.

The following table describes the parameters (assuming a "TRUE" value for use parameters):

PARAMETER	DESCRIPTION
BackPic	Name of the background picture file.
UseBack	Use the background picture
Picture1	Name of the picture file associated with trigger zone 1.
UseZone1	Use trigger zone 1
ShowOnStart	Display the picture associated with trigger zone 1 as soon as the protocol recording begins
HideOnPress	Hide the picture associated with trigger zone 1 as soon as the participant presses the pen on the tablet. **Caution**: this parameter will only be taken into account if "*ShowOnStart*" is activated.
UseZone2	Use trigger zone 2
Picture2	Name of the picture file associated with trigger zone 2.

Example

For the participant "*Toto*", the command
`RecNewPics&Usages(_P4,Bkgnd.bmp,TRUE,Capsela.bmp,TRUE,TRUE,`
`TRUE,car.bmp,FALSE):`
- the recorded data will be saved under the name "*toto_P4*";
- the background picture is "*Bkgnd.bmp*";
- the background picture will be displayed (*UseBack = TRUE)*;
- the picture file named "*Capsela.bmp*" is associated with trigger zone 1;
- trigger zone 1 will be used (*UseZone1 = TRUE*);
- the picture associated with trigger zone 1 will be displayed when the

- the picture associated with trigger zone 1 will be removed as soon as the participant presses the pen on the tablet to write or draw (*HideOnPress=TRUE*);
- the picture "*car.bmp*" is associated with trigger zone 2;
- trigger zone 2 will not be used (*UseZone2 = FALSE*).

```
SetPicsZones(x1Zone1,y1Zone1,x2Zone1,y2Zone1,
             x1Zone2,y1Zone2,x2Zone2,y2Zone2)
```

This command redefines the coordinates of trigger zones 1 and 2 on the tablet. These coordinates will replace the coordinates defined in the acquisition configuration panel of Eye and Pen (Simple tab). These new values will remain until the original values are restored by the "RestoreOriginalPicsZones" command (see below).

Replace *"X1Zone1"*, "Y*1Zone1*", "X*2Zone1*", "*Y2Zone1*" with the new coordinates you want to use for **trigger zone 1**.

Replace *"X1Zone2"*, "Y*1Zone2*", "X*2Zone2*", "*Y2Zone2*" with the new coordinates you want to use for **trigger zone 2**.

Reminder: coordinates are defined in tablet units (see p. 14).

Example
The command
```
SetPicsZones(27094,23480,30480,18203,27094,14626,30480,9434)
```
defines:
- the coordinates for trigger zone 1 :
  X1=*27094*, Y1=*23480*, X2=*30480*, Y2=*18203* ;

- the coordinates for trigger zone 2 :
  X1=*27094*, Y1=*14626*, X2=*30480*, Y2=*9434*.

```
RestoreOriginalPicsZones
```

This command restores the coordinates for trigger zones 1 and 2 that were originally defined in the acquisition configuration panel (Simple tab), before being redefined via the `SetPicsZones` command (see above).

```
RecNewAll(AddToParticipantName,BkPic,UseBack,
     Picture1,x1Zone1,y1Zone1,x2Zone1,y2Zone1,
     UseZone1,ShowOnStart,HideOnPress,
     Picture2,x1Zone2,y1Zone2,x2Zone2,y2Zone2,
     UseZone2,X1ZoneEnd,Y1ZoneEnd,X2ZoneEnd,Y2ZoneEnd)
```

This command launches the Simple acquisition paradigm, redefining all the parameters of the acquisition configuration panel (Simple tab).

> **Caution**: this command manages the opening and closing of the recording data files. This means that you do not have to use OpenRec and CloseRec.

Replace *"AddToParticipantName"* with a suffix that will be added to the recording file's name.

Replace *"x1Zone1"*, *"y1Zone1"*, *"x2Zone1"*, *"y2Zone1"* with new coordinates for trigger zone 1 (in tablet units, see p. 14).

Replace "x1zZne2", *"y1Zone2"*, *"x2Zone2"*, *"y2Zone2"* with new coordinates for trigger zone 2 (in tablet units).

Replace "x1ZoneEnd", *"y1ZoneEnd"*, *"x2ZoneEnd"*, *"y2ZoneEnd* with new coordinates for the "end" zone (in tablet units)

The other parameters can be given two values:
- "TRUE": activates the parameter;
- "FALSE": deactivates the parameter.

The following table describes the other parameters (assuming a "TRUE" value for use parameters):

PARAMETER	DESCRIPTION
UseBack	Use background picture.
BackPic	Name of the background picture file.
Picture1	Name of the picture file associated with trigger zone 1.
UseZone1	Use trigger zone 1.
ShowOnStart	Display the picture associated with trigger zone 1 as soon as the protocol recording begins.
HideOnPress	Hide the picture associated with trigger zone 1 as soon as the participant presses the pen on the tablet. **Caution**: this parameter will only be taken into account if "*ShowOnStart*" is activated.
UseZone2	Use trigger zone 2.
Picture2	Name of the picture file associated with trigger zone 2.

Example

For the participant "*Toto*", the command

```
RecNewAll(_P5,YellowBkgnd.bmp,TRUE,Capsela.bmp,27094,23480,
30480,18203,TRUE,TRUE,TRUE,car.bmp,27094,14626,30480,9434,
FALSE,12456,1845,14170,1235)
```
defines :

- the data that will be saved under the name "*toto_P5*" ("toto_P5.tab")
- the background picture is "*YellowBkgnd.bmp*";
- the background picture will be displayed (*UseBack = TRUE*);
- the picture file named "*Capsela.bmp*" is associated with trigger zone 1;
- the coordinates of trigger zone 1 on the tablet are :
  X1=27094, Y1=23480, X2=30480, Y2=1820
- trigger zone 1 will be used (*UseZone1 = TRUE*);
- "*Capsela.bmp*" will be displayed when the command starts (*ShowOnStart = TRUE*);
- the picture associated with trigger zone 1 will be removed as soon as the participant presses the pen on the tablet to write or draw (*HideOnPress=TRUE*);
- the picture "*car.bmp*" is associated with trigger zone 2
- the coordinates of trigger zone 2 on the tablet are:
  X1=27094, Y1=14626, X2=30480, Y2=9434;
- trigger zone 2 will not be used (*UseZone2 = FALSE*);
- the coordinates of the "end" zone on the tablet are:
  X1=12456, Y1=1845, X2=14170, Y2=1235.

Sets the maximum display duration for pictures associated with tablet zones

SetRecStandardMaxDisplay(MaxDurationForPicture1, MaxDurationForPicture2)

This command defines the maximum display duration for the pictures associated with trigger zones 1 and 2 of the Simple acquisition paradigm (`RecStandard`).

Example

The command `SetRecStandardMaxDisplay(1000,2000)` limits the duration of the picture display to 1 second (1,000 milliseconds) when the participant presses the pen in zone 1 and to 2 seconds when the pen is pressed in zone 2.

Special value: -1 (unlimited duration). This is the default value when no value has been explicitly defined.

Sets a maximum time for production

SetMaxWritingDuration(MaxDuration)

This command does not suspend script execution, but does set the maximum time (in milliseconds) during which the participant can write (for commands `WaitForTabZoneAt`, `WaitForTabZones` and `RecStandard`).

Replace `MaxDuration` with a duration expressed in milliseconds.

Example

The command Set`MaxWritingDuration(360000)` limits the duration to 6 minutes (6*60*1,000). If the participant has not pressed the pen in one of the defined zones (or end zone for `RecStandard`) by the time this period has elapsed, the command terminates "on its own" and script execution continues.

Special value: -1 (unlimited duration). This is the default value when no value has been explicitly defined.

IV.   NETSYNC
(*File/Acquisition/NetSync* Menu)

IV.1. Presentation

Script-based acquisition can be managed in a network environment, thanks to the NetSync module.
This tool is designed to save time and enhance data security: instead of individual test sessions, it allows group sessions to be held and enables the entire volume of data that is created to be processed, all the while limiting risks of data loss.

The concept is based on a Client-Server architecture. It is built around two applications: a "controller" application, the so-called NetSync module, executed on the computer acting as a server (known as the "master host") and a client application undertaking data acquisition on the "Client host" computer.
The script that is to be executed on each individual "Client" host is selected from the "master host" application.
The pace at which the test is administered on client hosts can be controlled from the master host through a "client pause/authorization to continue from master" mechanism, thanks to a specific command inserted into the acquisition script.
When the scripted acquisition is over, data can be copied from each client host to the master host, where they are stored and organized according to the date and time of the session, the participant's name and the client host's name, thus preventing data from being overwritten.

NetSync is designed to act as a network "synchronizer", managing collective experiments. Client hosts' stimuli and scripts can be updated from the master host's own stimuli and scripts and their data can be copied to the master host when the acquisition session has ended.
Data acquisition is based on the client's local configuration (software and hardware).

IV.2. Client

The client side application is simply the Eye and Pen launched with the command line parameter "NETSYNC".
Thus, to start a computer as a netsync client, launch "EP.EXE NETSYNC".
You may also use the "Netsync client" shortcut found in the "Eye and Pen 2" program group in the Windows Start menu.

There is no order for launching applications: if the client is launched before the master it will keep on trying to connect until either it succeeds or you stop it.

There is no check for dongle presence with a NetSync client.
You may therefore use as many Eye and Pen installations as the master host is able to accept (this number is defined by the master host dongle).
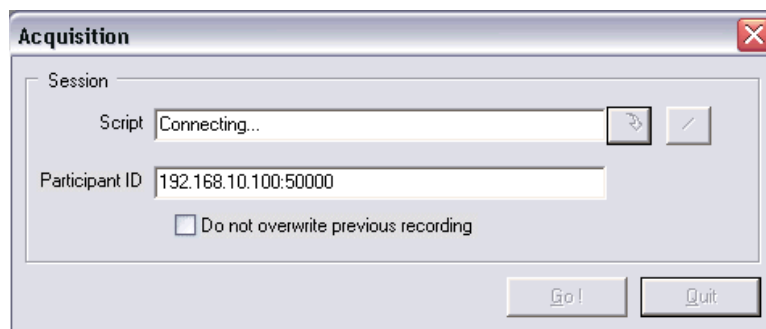
Figure 39: NetSync client connecting.

After the client application has been launched, it will try to establish a connection with the server (master).

Once this connection has been established, if the *Stimuli* and *Scripts* folder update option has been selected (*Configuration / Acquisition / NetSync*, p. 43), the client will download any missing or very recent files.

This update feature may be used to distribute scripts and stimuli to client hosts without having to manually copy theses files to each host.

Next, the client sends a list of the scripts (*.txt) contained in its *Scripts* folder to the server (defined in the *Configuration / Acquisition / Scripts*, *Scripts* folder option).

The client then waits for the server to send information about the session (session name, participant identifier) and the script to be executed.

Note:

> The session name will be used to create a subfolder in the client's data folder (as defined in the *Configuration / Acquisition / Scripts, Data Folder* option).

Once script execution has started on the client host, the latter may send a message to the master host notifying it that it is waiting for an authorization to continue (via the script command `WaitForNetSync`), or send a "common" information message to the master (via the command `SendMessageToNetSync`) whose content will be displayed in the "Status" column (see Fig. 40).

As soon as the client has finished its scripted acquisition process, it will automatically send an end message to the server.

The client then waits either for a "disconnect" message from the server, or else for an order to send a copy of its acquisition session data files (these files will remain in the client host's data folder).

Once the client has received the "end" message, it will start waiting for a new acquisition session.

Note: if a problem arises (faulty network connection, etc.), the "connecting to server" procedure can be aborted by pressing the Esc key on the client's keyboard or by clicking on the X in the upper right-hand corner of the dialog box. The software will immediately stop running.

IV.3. Master

The master application (the server) is launched from the "*File / Acquisition / NetSync*" menu in Eye and Pen.

There is no order for launching applications: if the client is started up before the server it will

keep on trying to connect until either the server (Master) accepts the connection or the application is closed.

Once it has been launched, the master application can accept any client at any time.

The server can accept a maximum of 25 clients. Please note that the efficiency of the network connections is governed by the master's performance and the quality of the network.
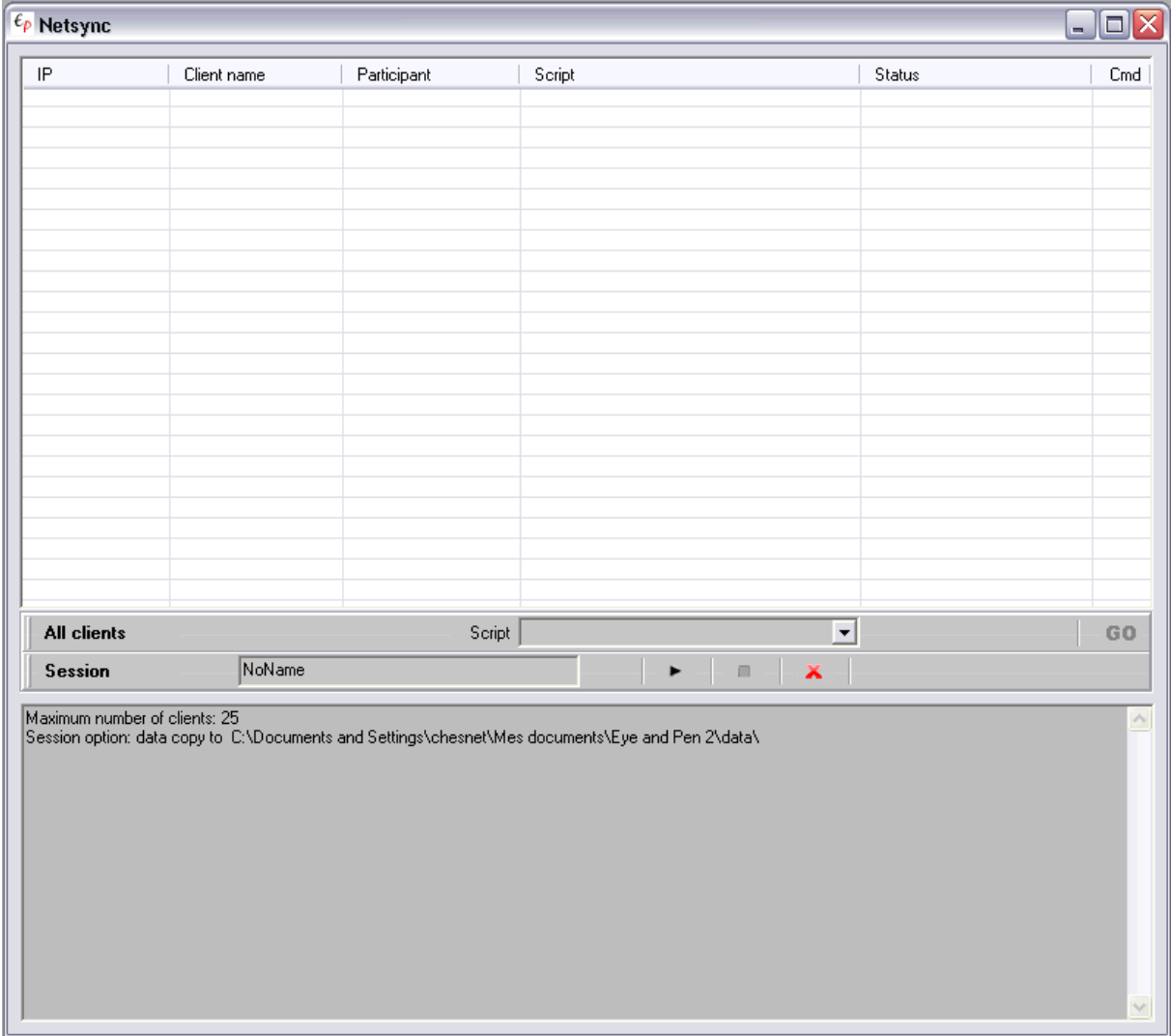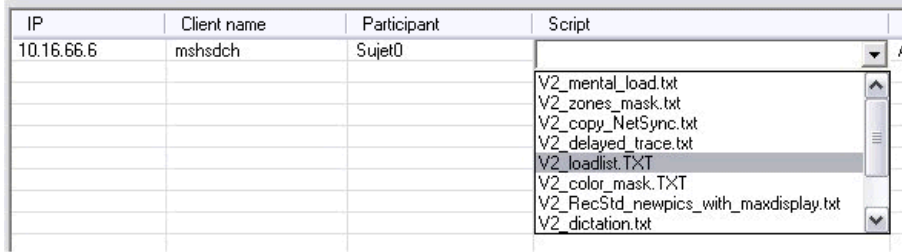
Figure 40: NetSync interface.

When a client is connected, information is automatically added to the "dashboard" in a new line. This information is set out below:

LABEL	DESCRIPTION
IP	IP (network) address of the client host. This data is informative and cannot, therefore, be edited by the master host's application.
Client name	Client host's name on the network. This data is informative and cannot, therefore, be edited in the master host's dashboard.
Participant	To edit the participant's identifier, double-click on the dashboard cell or press the F2 key of the keyboard on the master host.
	<u>Caution</u>: the characters ' \ " / : * ? < > | % are not allowed. If you try to use one of them, a dialog box will inform you that it is not allowed (the

	participant's name will be used to structure filenames).
Script	Filename of the script to be executed on the client host.
	To select the script filename, click on the cell and select the name from the "Script" scrolling list that appears. This list contains all the scripts found in the client host's *Scripts* folder (this list is sent to the master when establishing the connection).

Figure 41: Selecting a script for a client.

Status	Shows the status message returned by the client host (predefined messages).

Label	Description
Waiting for session	The client is waiting for the session information.
Session received	The client acknowledges that it has received the session information.
Waiting for Go	The client is waiting for an authorization from the server (click on the "Go" button at the end of the line or on the "Go" button in the "All clients" toolbar.
Working	The client is executing its script.
Job completed	Script execution is terminated.
Copy started	Data copy from the client has started.
Copy completed	Data copy from the client is over.
Updating	The client is downloading scripts and stimuli.

	This column can also display messages sent by the client via the `SendMessageToNetSync` command.
Cmd	Contains a "Go" button that is disabled (shaded) during the connection stage. This button is enabled when the client sends the `WaitForNetSync` command.

If a client breaks its connection, its line will remain in the dashboard, but its information will be erased (Participant, Script, Status).

A client can be deleted or added to the session that is currently running via a mouse right-click on its line on the dashboard (contextual menu).

Below the dashboard you will find two toolbars.
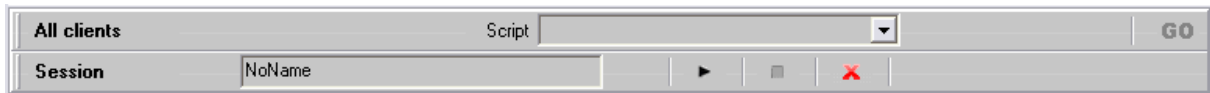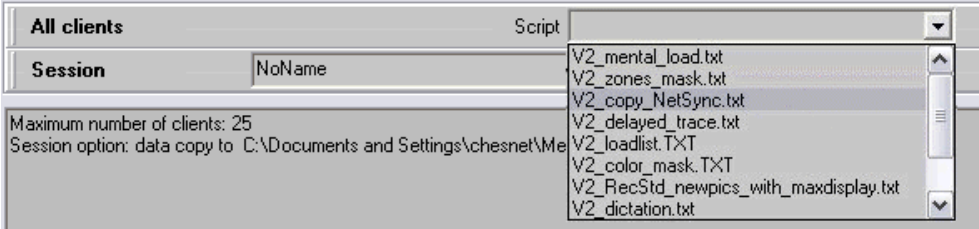
Figure 42: NetSync toolbars.

The first toolbar allows you to manage the whole set of clients at once. The following table describes its components:

LABEL	DESCRIPTION
All clients	This list contains the script names that are common to all the clients. When a script is selected from this list, it is assigned to each of the clients (the "Script" field in the dashboard will be updated with this name). *Figure 43: Selecting a script for all the clients.*
GO	Allows you to send a "start" signal to all the client hosts. This message is meant to be used when clients have requested permission to continue (the "Status" column of the dashboard displays the "Waiting for Go" message). This button is enabled as soon as the acquisition session has started.

The second toolbar is dedicated to managing the acquisition session.

LABEL	DESCRIPTION
`Session`	Name you give to the session. The date, hour and seconds will automatically be added to the name.
`Start session`	When you click on this button, it triggers the sending of information required by each client to start an acquisition (session "name" field content, current time and date, "Participant" and "Script" names for the client). The application checks that a script has been selected for each client. If not, a dialog box lists those clients who do not meet this condition and stops the process, allowing the situation to be rectified. As soon as the session has started, the "Session end" button is enabled.
`End session`	This button allows you to trigger the ending of the acquisition session, for example copying data from the client to the server (if this option has been selected in the *File / Configuration / Acquisition / NetSync / Transfer...* menu). Once these operations have been completed, the "Session start" button is enabled again, allowing you to start a new acquisition session with another script, for example.

✕ Close clients	This button allows you to disconnect all NetSync clients. Each client application will therefore be shut down after disconnection).

The panel at the bottom of the NetSync dashboard is used to display messages relating to the server's work (transfer option, data transmission, etc.). When data are uploaded by clients, the name of each copied file is displayed in this panel.

Log file

On the Master host, the parameters of each acquisition session are written in the file named "NetSessions.log" (in the Eye and Pen *Data* folder).

If the client data copy has been selected (see NetSync configuration, p. 43), the session-specific data will be recorded in a file named *<SessionName>*.log, in the folder containing the acquisition data.

A few recommendations

✓ Client host identification is based on clients' IP addresses. In Windows Vista/7, the "Network Discovery" service must be enabled for this mechanism to succeed[3].

✓ Network communication may be blocked by the Windows Firewall, so make sure that you have disabled it for the duration of the acquisition.

✓ When managing an acquisition with NetSync, everything that can use up resources (and, above all, everything that can use up network resources), is a potential hindrance to data acquistion. We strongly advise you to "clean" acquisition computers, removing software such as Messenger, AIM, NetMeeting, Chat, etc.

✓ For the duration of data acquisition, disable antiviruses, connections and network services not required by the experiment (Bluetooth, etc.).

✓ Disable everything you can, including all the "Quick Start" programs of software such as Adobe Reader, OpenOffice and Microsoft Office, together with the automatic updates.

A safe and simple method is to set up your own private network that is not connected to the Internet. In the Appendix (p. 187), you will find an example of a WiFi private network we used in a classroom.

---

[3] http://windows.microsoft.com/en-us/windows-vista/What-is-network-discovery
http://windows.microsoft.com/en-us/windows-vista/Enable-or-disable-network-discovery

V.   RETROSPECTIVE COMMENTS
(*Display / Retrospective* menu to show/hide the tool)

This tool is a sort of Dictaphone whose recordings are kept in synchrony with the protocol analysis. It can be regarded as an acquisition tool (e.g. you can ask the participant to comment on his or her own production afterwards, or about that of someone else), but also as a tool to help analysis (e.g. making vocal annotations about the coding of a protocol).
To be able to use it, you must own an audio recording device (e.g. a microphone) and playing device (e.g. speakers or headphones).

A *retrospective comment* is made up of an audio recording, a start timestamp and a name. The "*Retrospective*" tool allows you to:
- record/listen to a recording;
- directly "jump" to the start time (insertion point) of a recording in the protocol.
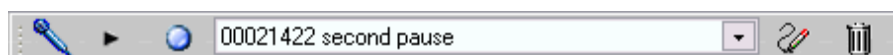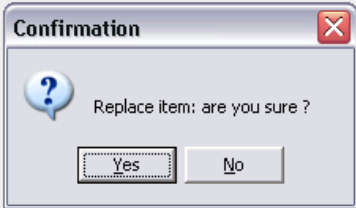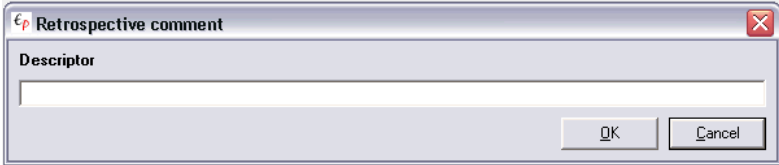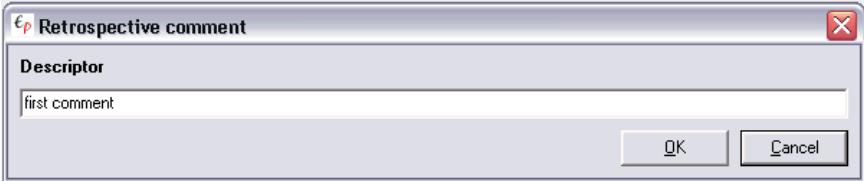
Figure 44: Toolbar for recording retrospective comments

The recording is saved in an audio Wave file format (.WAV) and is "linked" to the current tablet data timestamp (in the protocol) for the start of the comment recording. For example, if a recording was started at timestamp 2,154 milliseconds in the protocol ACI_108 (the timestamp can be seen in the "clock" tool), a field named ACI_108.00002145.wav will be created in the data folder.
The list of retrospective comments is saved in a (*<participant>*.RTR) file when the analysis is saved.

The following table explains the function of the tools in the toolbar:

ICON	DESCRIPTION
	Record. Selects the current protocol timestamp (time elapsed since the start of data acquisition as displayed by the "clock" tool) as a link for the recording and starts the audio recording. If the current timestamp of the protocol is that of a previous audio recording a dialog box will warn you about it and prompt for overwriting.
	Listen to the selected comment.
	Stop. This button is red while listening to or recording a comment. If you stop a recording, a dialog box will prompt for a descriptor (a "name" or label). *Figure 45: Describing a retrospective comment.* Type the comment's label in the "***Descriptor***" field, then click on the "OK" button to validate.

	The comment's timestamp and descriptor will then be displayed to the left of the ▼ button in the toolbar.
✒	**Edit.** To change a comment's descriptor (its "name"), select this comment (see button ▼, below) then click on the ✒ button. The comment's descriptor dialog box will appear: Ɛₚ Retrospective comment ✕ **Descriptor** first comment OK Cancel *Figure 46: Modifying a comment's label.* Modify the comment's name (the dialog box can be resized). Click on the "*OK*" button to validate.
🗑	**Delete.** To delete a comment, select its descriptor (see button ▼, below), then click on the trashcan icon. When a "retrospective comments" list entry is deleted, the associated Wave file is sent to the Windows recycle bin.
▼	**Select**. When you click on this icon, a list scrolls down and displays the existing comments' labels. Click on the comment label of your choice to jump to the start of the selected comment. Important: comment labels are listed in order of creation, starting from the top.

CHAPTER 3: data analysis

I.   INTRODUCTION

Data analysis allows you to sort, code and export recorded data (into text files).

In Eye and Pen, data analysis is based on the principle of the VCR: things are viewed as though the participant were being "filmed" during production. You can then watch the resulting movie in a variety of ways. To do this, different tools allow you to "navigate" through the data, moving forward or backwards with self-defined filtering parameters.

As soon as you launch the Simple or scripted acquisition mode, tablet and eye tracking data are recorded and "stamped" with a common time baseline (in milliseconds).

To make data geometrically compatible, the coordinates of the eye tracking data are converted into the same coordinates system as the tablet data. The tablet is used as a reference.

Example

If the tablet coordinates are in a (0, 0, 30240, 30240) frame and the "eye" coordinates are in a (0, 0, 1024, 768) frame, the "eye" coordinates are scaled to the (0, 0, 30240, 30240) frame so that both sets of information can be stacked on the same surface.

Important:
If the sheet of paper slips on the tablet, or the eye tracker helmet moves on the participant's head, eye-tracking and tablet data will not be spatially accurate. Thus, **some eye tracking data may be outside the calibrated area instead of inside, where it should be**. To help solve this kind of problem, "Eye and Pen" has a tool that can "shift" an entire set of data (see page 136).

Hint:
To avoid pointless extra work, we recommend that you initially apply every **overall filter** in order to reduce the total amount of data (shift layers, fixation building, out-of-field data cleaning etc.). Next, apply the **automatic coding filters** (Areas Of Interest, etc.) and only then begin the **qualitative and manual analyses** (coding, etc.).
Here is an example of what should be avoided: you have coded some pauses "by hand" and you then apply an automatic coding. The codes of these pauses may change, even if you do not want this to happen.
It is important to carry out these operations in the right order.

The analysis may be closed at any time, by clicking on the "*File*" menu, then selecting "*Close analysis*".

II. START AN ANALYSIS

When analysing data for the first time, you may want to configure the data analysis, see "Data analysis configuration" in <u>chapter 1</u>.

To start a data analysis, you can:

- **Start a new analysis:** click on the "*File*" menu, then select "*New analysis*". A dialog box will be displayed so that you can select the tablet data file (file with a ".tab" extension) you want to analyze. Click on the "*Open*" button;

- **Reload a previous analysis:** click on the "*File*" menu, then select "*Re-open*" and "*Open*". A dialog box will let you select the analysis data file (file with a ".twk" extension) you want to work on again. Click on the "*Open*" button;

- **Re-open an acquisition session:** click on the "*File*" menu, and then select the "*Session*" submenu. A dialog box shows up. Then, one can select the session "log" file (the filename extension is « .LOG ») that you intend to explore. Next, click on the "Open" button;

- **Import tablet data from G-studio files** (written production recording software for the MS-DOS system, also based on digitizing tablets). Click on the "*File*" menu, select "*Imports*" and "*G-studio*". A dialog box will be displayed to allow you to select the file you want to import into Eye & Pen (file with a "._" extension). Click on the "*Open*" button to start the analysis.

III. RECORDING SESSION (LOG)

For each "Script" acquisition session, a file named (<Data Dir.><Participant Id.>**.LOG**) is created, incorporating all the main interactive events involving the participant (displays, recordings, participant's responses, etc.), with their corresponding times (milliseconds). Timing starts at the very beginning of the acquisition session (time zero).
This information is shown as a tree view and sorted by time (the oldest event at the top). In the following example, the script was loaded at 10 milliseconds, a picture was displayed at 45 milliseconds, etc. (for code meanings, see p. 196).
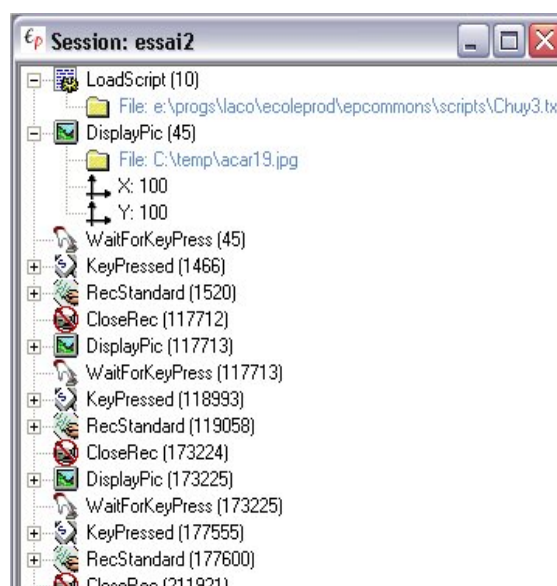
Figure 47: Acquisition session.

About events and timestamps…

The time indicated is the event start time. For example, when a picture is displayed, the timestamp is the very point at which the display procedure started, not the time when the picture became fully visible to the participant.

Elements tagged "File" can be clicked on and then either opened in a Windows application or else in Eye and Pen, if it is a picture or a protocol recorded using the `RecStandard` command.
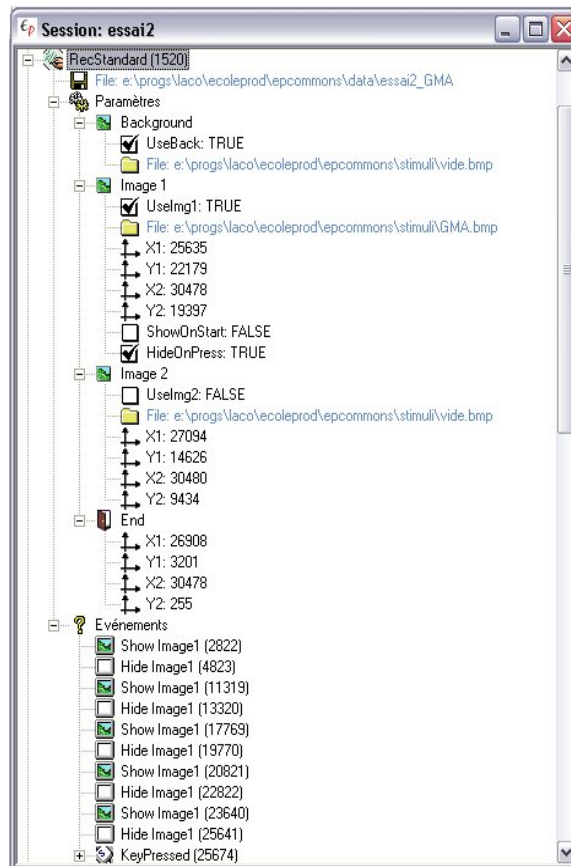
Figure 48: A "RecStandard" recording session log.

When opening a `RecStandard` via the "Session" menu (click on the floppy icon), the associated parameters recorded in the log file are loaded (background, pictures 1 and 2, and end zone). During a `RecStandard` recording, unexpected events are also recorded (e.g. a keyboard keypress or the end of a sound) and are treated as sub-events of `RecStandard`.

The session log indicates the true display coordinates of pictures and videos. For example, if the script includes the command `DisplayPic`("toto.bmp",2000,-1,-1), i.e. display toto.bmp for 2,000 milliseconds centered in the display window, the log file will show the exact coordinates of the upper left-hand corner of the picture in the display.

The log file is a "text-only" file type, so you can load it in various other types of software (text editor, word processor, spreadsheet, databases, etc.).

IV. INFORMATION AVAILABLE AT ANY TIME

In "analysis" mode, the screen is composed of a number of elements bar:
- 7 navigation tools:
  - ➢ from event to event;
  - ➢ from pause to pause;
  - ➢ from fixation to fixation;
  - ➢ by tablet data code;
  - ➢ by eye data code;
  - ➢ with the distance between pen and gaze locations;
  - ➢ a stopwatch (clock);
- a tool for creating sequences in the protocol;
- a tool to record the retrospective verbalisations;
- a zoom;
- a key press recorder;
- a tool for coding data;
- an information bar (status bar);
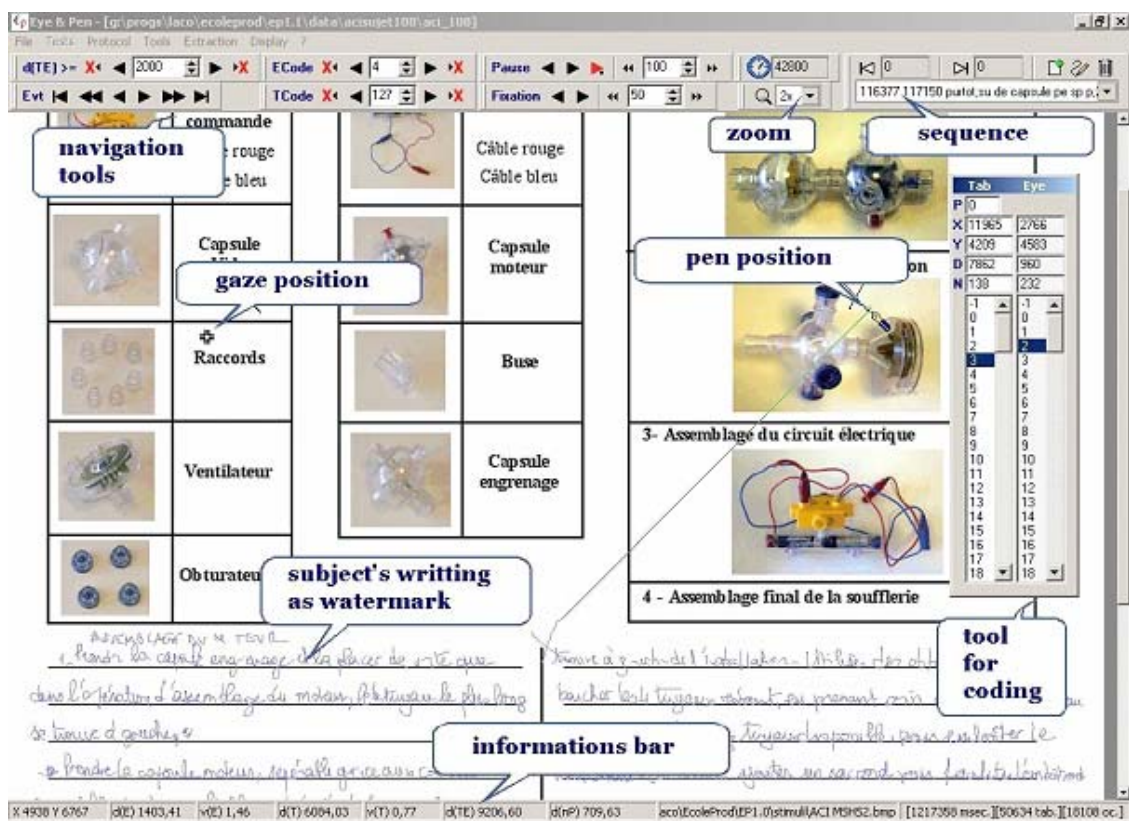
Figure 49: Data analysis screen.

These tools can be shown or hidden via the "Display" menu:
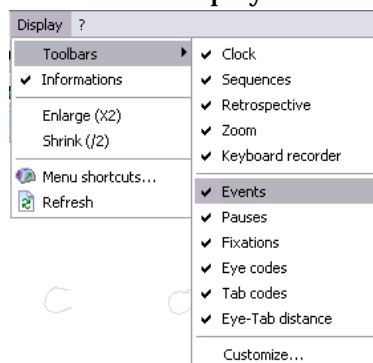
Figure 50: Menu to display tools.

<u>Pause representation</u>

When the pen is "Up" (pausing), a circle indicates the point where the pen left the tablet and a line connects this circle to the place where the pen subsequently "lands" (see *File/Configuration/Analysis* menu, "*Analysis*" tab to select "pause color")

Figure 51: Representation of a pause.

IV.1. A tool for coding
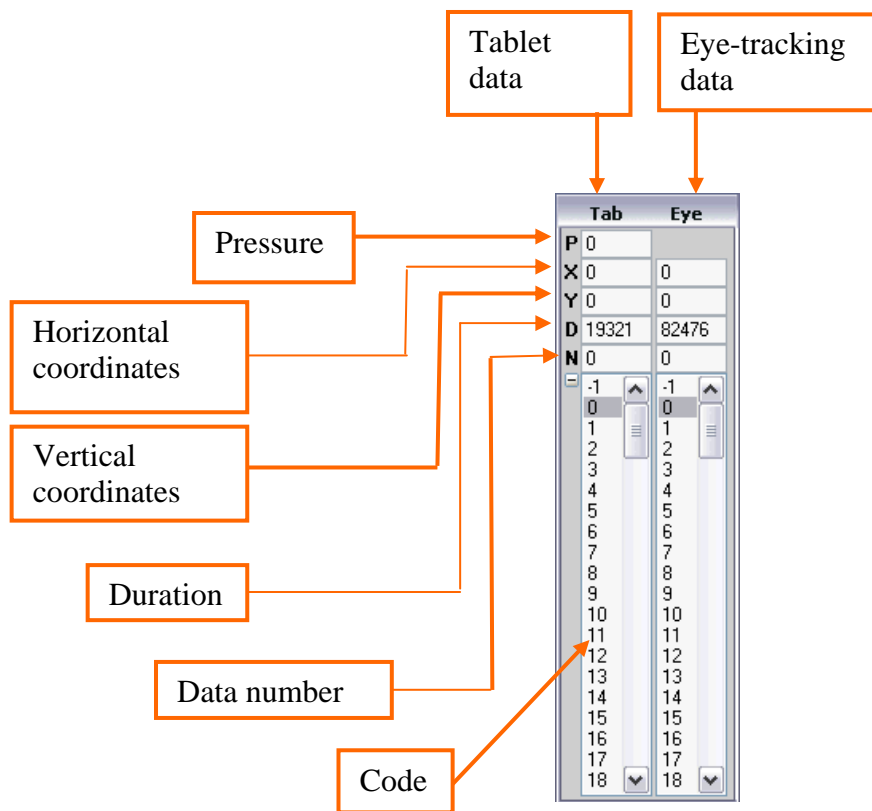
Figure 52: The coding tool.

While navigating through the protocol, the coding tool will show details of the current tablet and eye-tracking data. The code is a digit value (it is up to you to give it meaning).

LABEL	DESCRIPTION
P	Pressure exerted with the pen on the tablet.
⚠	When the gaze position is beyond the calibrated area, a warning signal is displayed to the right of the pen pressure field, in the "Eye" data column.

X	Coordinate of the data (tablet or eye) on the horizontal axis.
Y	Coordinate of the data (tablet or eye) on the vertical axis.
D	Duration (in milliseconds) of the event, defined as the time that elapses between the current data and the following data of the same type (tablet or eye).
N	Data number (numbering from zero upwards). Each type of data is numbered separately.

You can reduce or enlarge the tool by double-clicking on its title bar. You can also move it.

IV.2. Information bar (status bar)

(*Display/Information* menu to show/hide this tool)

This tool provides additional information about the current data.

X 634 Y 1061	d(E) 4558,91	v(E) 2279,46	d(T) 1556,92	v(T) 0,38	d(TE) 5779,00	d(nP)	E:\PROGS\laco\EcoleProd\EP1.1\point.bmp	[40193 msec][1442 tab][81 eye]
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**

Figure 53: the information bar.

This information is described from left to right.

LABEL		DESCRIPTION
(1)	**X** **Y**	Mouse cursor's horizontal and vertical coordinates, expressed in tablet units. It may help you to check for the coordinates of a particular place in the protocol.
(2)	**d(E)**	Euclidian distance between current eye data (E stands for Eye) and the position of following eye data (in tablet units).
(3)	**v(E)**	Speed of movement between current eye data position and following eye position (in tablet units per millisecond).
(4)	**d(T)**	Euclidian distance between current eye data (T stands for Tablet) and the position of the following eye data (in tablet units).
(5)	**v(T)**	Speed of movement between current pen (Tab) data position and following pen position (in tablet units per millisecond)
(6)	**d(TE)**	Euclidian distance between current tablet data position and current eye data position (in tablet units).
(7)	**d(nP)**	Tracing distance (pen movements) between the end of the last "up" pause and the following "up" pause (in tablet units). This information can help you anticipate the next pause.

(8) c:\…	Path and name of the background picture. Click on it to select a new one.
(9) **[msec]** **[tab]** **[eye]**	**[** Total duration of protocol (in milliseconds) **]** **[** Total amount of tablet data ("tab.") **]** **[** Total amount of eye-tracking data ("oc.") **]**

IV.3. History

(*Protocol/History* menu)
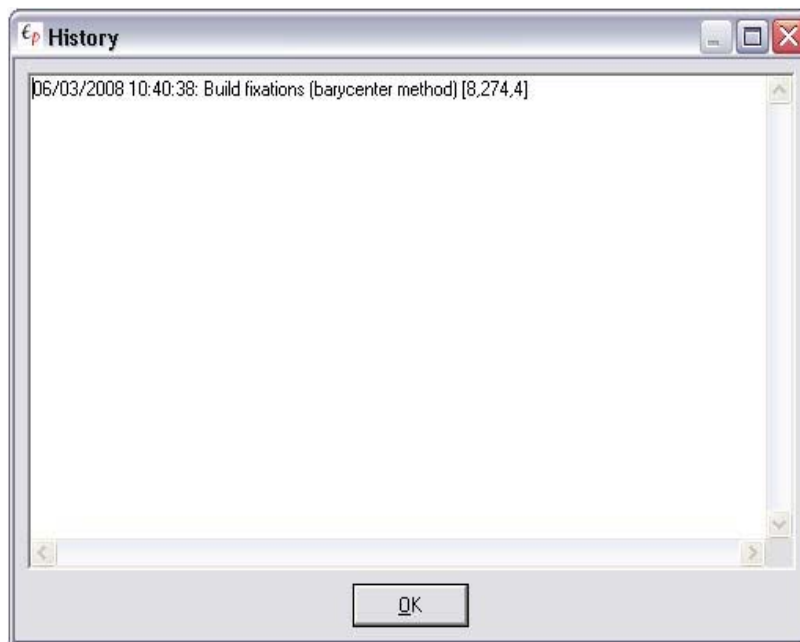
Figure 54: History dialog box listing actions.

This menu displays the list of processes that have been applied to the protocol in order to reduce the total amount of data, such as building fixations or the aggregation of successive data with the same code.

When an action is canceled, the corresponding entry is removed from this list.

V. NAVIGATION

V.1. Introduction

You can navigate through the data in two ways:
- "spatially", by clicking with the right mouse button first on the place in the protocol that you would like to jump to (participant's writing on the tablet), then on the "*Jump here*" contextual menu that appears (see below).
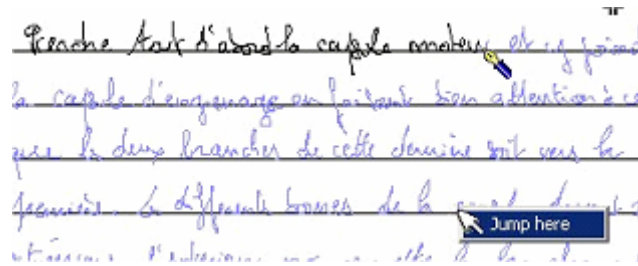
Figure 55: "Spatial" navigation.

Eye and Pen searches for the nearest item of data in a 5-pixel square around the mouse cursor position. When it has been found, the protocol is "played" until it reaches this item.

- searching for a particular category of event:
  - from one item of data to the next;
  - from one pause to the next;
  - from one fixation to the next;
  - searching for the next item of "eye" or "pen" data with a particular code value;
  - searching for the next point where the distance between "eye" and "pen" data meets a given criterion.

The "***Display***" **menu** allows you to show or hide the **navigation toolbars** located under the menu.
These tools can be moved using "*drag'n'drop*":

STEP	DESCRIPTION	
1	Click with the left mouse button on the vertical "line" left of the toolbar (to the left of its label)	
2	Keep the left mouse button pressed down	
3	Move the toolbar to the desired location.	
4	Release the mouse button.	

Important:

➢ the "**shift + T**" shortcut re-centers the protocol to make the pen cursor visible, i.e. on the screen ;

➢ the "**shift + E**" shortcut re-centers the protocol to make the eye cursor visible, i.e. on the screen.
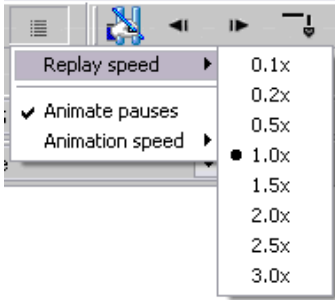
V.2. Navigating from one event to another

(*Display/Events/Navigation* menu allows you to show/hide this tool)
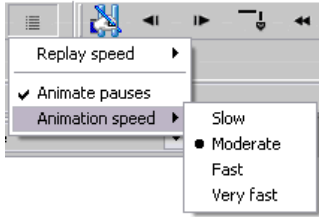
This tool allows you to navigate within the protocol, from one event to another, through the whole set of data (tablet and eye-tracking). An event is either a movement or an immobilization of the pen or eye. An item of data is understood to be the content of an event: its coordinates, its timestamp, etc.

Figure 56: Navigating from one event to another.

ICON	DESCRIPTION
⏮	Jump to the beginning of the protocol.
◀ᴵ	Previous event.
◀◀	Move continuously back through the events (does not respect events duration).
▶	Move continuously forward through the events, respecting events duration. Replay speed can be changed (see options).
▶▶	Move continuously forward through the events (does not respect events duration).
ᴵ▶	Next event.
▶ᴵ	Jump to the end of the protocol.
☰	Replay options. A menu allows you to set replay speed (from 0,1 to 4 times the normal speed). *Figure 57: Set the protocol replay speed.* Protocol replay may be animated, during pauses. A circle (centered at pause beginning location) will grow as time goes by. Its growth speed may be selected amongst four values ranging from "slow" to "very fast".

Figure 58: animate pauses when replaying a protocol.

This option menu may also be activated with a mouse right-click on the "movie" icon.

	When replaying events in a continuous way, the clicked button becomes a "stop" button: click it to stop the replay.

Figure 59: Stop protocol replay.

V.3. Navigating from one pause to another
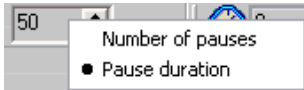
(*Display/Pauses navigation* menu to show/hide this tool)
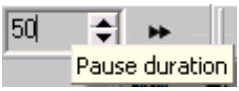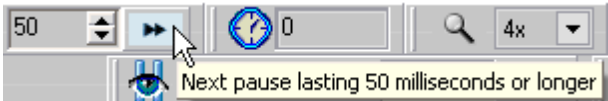
This tool helps you to navigate from one pause to another, for a greater duration than the defined threshold (see <u>Thresholds</u>)**.** Its icon comprises a hand holding a pen and the pause symbol used in audio/video players (a vertical double bar).

This tool features fast-forward and rewind functions, either per number of pauses or per pause duration (greater or equal to a defined value).

Figure 60: Navigating through pauses.

ICON	DESCRIPTION
	Previous pause (up or down).
	Next pause (up or down).
	Go to the end of the current pause.
	Step back by "***n***" pauses (up or down) or jump to the previous pause whose duration is longer than the given value.
	Number of pauses (or duration) for the "backward/forward by ***n*** pauses" function. A mouse right-click opens a contextual menu allowing you to choose what the content of this field stands for: number or duration. *Figure 61: Choose navigation criterion: duration or number of pauses.* <u>Hint</u>: when the mouse hovers above the "value" field, a tooltip shows up to

	indicate the current meaning of the value:
	For pause duration, the smallest value is the smaller of the two sorts of pauses: up or down.
▸▸	Step forward by "***n***" pauses (up or down) or go to the next pause whose duration is longer than the given value. <u>Hint</u>: when the mouse hovers the "value" field, a tooltip shows up to indicate what is the current meaning of the value:

V.4. Navigating from one fixation to another

(*Display/Fixations navigation* menu to show/hide this tool)

This tool helps you to navigate from one fixation to another, with a greater duration than the defined threshold (see [Thresholds](#))**.** Its icon comprises an eye and the pause symbol in audio/video players (a vertical double bar).
This tool features fast-forward and rewind functions, either per number of fixations or per fixations durations (greater or equal to a defined value).
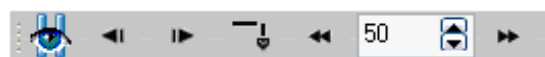
Figure 62: Navigating through fixations.

ICON	DESCRIPTION
◂ı	Previous fixation.
ı▸	Next fixation.
⊥̬	Go to the end of the current fixation.
◂◂	Step back by "***n***" fixations or jump to the previous fixation whose duration is longer than the given value.
100 ⬍	Number of fixations for the "backward/forward by ***n*** fixations" functions or duration. A mouse right-click opens a contextual menu allowing to choose what the content of this field stands for: number or duration. *Figure 63: Choose navigation criterion: duration or fixation number* <u>Hint</u>: when the mouse hovers the "value" field, a tooltip shows up to indicate

	what is the current meaning of the value:
	For fixation duration, the smallest value is the value of the fixation threshold.
▸▸	Move "***n***" fixations forwards or go to the next fixation whose duration is longer than the given value. Hint: when the mouse hovers the "value" field, a tooltip shows up to indicate what is the current meaning of the value:

V.5. Navigating through "Eye" data filtered by code

(*Display/Eye codes Navigation* menu to show/hide this tool)

This tool helps you to navigate through the eye tracking data ("Eye"), filtered by a code you have selected (see Data coding). This enables you to jump from one item of Eye data with the required code to another. Its icon comprises an eye with a sharp symbol (i.e. numbering) on top of it.
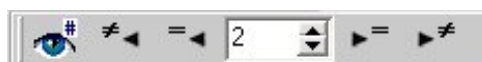
Figure 64: Navigating with Eye data codes.

ICON	DESCRIPTION
≠◂	Go back to the previous "Eye" data item without the required code value.
=◂	Go to the previous Tab data item with the required code.
2 ⬍	Code value.
▸=	Go to the next "Eye" data item with the required code.
▸≠	Go to the next "Eye" data item without the required code value.

VI.6. Navigating through "Tab" data filtered by code

(*Display/Tab codes navigation* menu to show/hide this tool)

This tool helps you to navigate through the tablet data ("Tab"), filtered by a code you have selected (see Data coding). This means that you can jump from one item of tablet data showing the required code to another. Its icon comprises a hand with a sharp symbol (i.e.

numbering) on top of it.

Figure 65: Navigating with tablet data codes.

ICON	DESCRIPTION
≠◄	Go back to the previous "Tab" data item without the required code value.
=◄	Go to the previous "Tab" data item with the required code.
110	Code value.
►=	Go to the next "Tab" data item with the required code.
►≠	Go to the next "Tab" data item without the required code value.

V.7.   Navigating through data filtered by gaze-pen distance

(*Display/Eye-Tab distance navigation* menu to show/hide this tool)

This tool allows you to navigate through the data on the basis of the (Euclidian) distance between the simultaneous gaze and pen positions (in tablet units).

Figure 66: Navigating according to the distance between gaze and pen positions.

ICON	DESCRIPTION
‹◄	Jump back to the previous point where the distance was below the criterion.
≻◄	Go back to the previous point (time) where the distance was equal to or above the criterion.
2000	Distance criterion (in tablet units).
►≻	Go to the next point where the distance was equal to or above the criterion.
►‹	Jump to the next point where the distance was below the criterion.

V.8.   Navigating through time with the "clock"

(*Display/Clock* menu to show/hide this tool)

As soon as data recording starts, each item of data is "time-stamped" with millisecond timing. Time "0" (zero) is the point where the recording started.

This tool allows you to navigate through the data, according to the time.
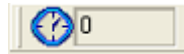
Figure 67: The clock.

The time (in milliseconds) that has elapsed since the beginning of the recording is displayed to the right of the "clock" icon. When you click on the icon, the dialog box below appears:
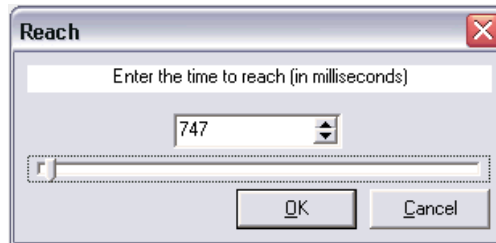
Figure 68: Dialog box for selecting a particular "time" in the protocol.

There are two ways of reaching a particular point in the data:
- key in (or select with the up/down arrow) the desired timing in the box;
- slide the cursor to select the time you want.

VI. THRESHOLDS

(*Protocol/Thresholds* menu)

Thresholds are used as filters. They are involved in navigation and calculations, including data outputs and previews.
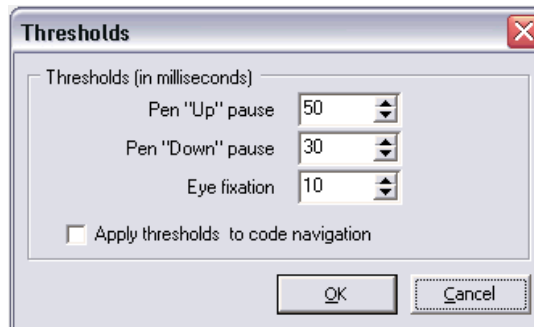
Figure 69: Threshold dialog box.

LABEL	DESCRIPTION
Pen "Up" pause	A pen event lasting longer than this value (with zero pressure) is considered to be a pause.
Pen "Down" pause	A pen event lasting longer than this value (with a pressure above zero) is considered to be a pause.
Eye fixation	An eye event lasting longer than this value is considered to be a fixation.
Apply thresholds to code navigation	If this option is ticked, the above thresholds will be applied as a secondary filter when using the "*By Tab code*" and "*By Eye code*" navigation tools (see page 123), i.e. an item of data must meet a double criterion (code + duration) if it is to be selected.

Minimum threshold values

For events only defined by their duration (down pauses or fixations), the minimum value of their threshold is the duration of two samples. Considering shorter durations may lead to accept each sample as a possible fixation or pause.

Expressed in milliseconds, the minimum value for thresholds is determined by the formula 2*(1,000/Sampling rate). For example, with a 200Hz sampling tablet, i.e. sampling 200 pen positions per second (=1,000 milliseconds), the minimum threshold will be: 2*(1,000/200)= 10 milliseconds.

As the "loss" of one or two consecutive data samples may occur during the acquisition or transmission of the data to the program, we strongly suggest a minimum value equal to the duration of 3 samples.

Hint: to view/edit all pauses (or fixations), set the minimum corresponding threshold.

VII.   BUILDING FIXATIONS

Eye fixations are a construction and do not "naturally" exist. Building a fixation basically consists in grouping data as though they represented the same event (the eye remaining at a given location for a certain amount of time). Many methods exists to calculate fixations, each one has its drawbacks and arouses controversy (cf. Bibliography). Both methods implemented in Eye and Pen are of the "position variance" (geostatic) type.

Caution: take care not to build more than one fixation at a time in the same protocol, as this method is data destructive. Once a fixation is determined, all the samples within it are replaced by the fixation centroid. A "second go" may have the effect of clustering fixations whose centers have moved closer to each other. When in doubt, consult the history (*Protocol/History* menu)

Note: fixation thresholds (see Thresholds, p. 125) allow you to filter fixations according to their duration. Fixations below this threshold will not be taken into account for navigating or editing.

VII.1. Barycenter method

(*Tools/Build fixations (barycenter method)* menu)

Method:

An initial item of data is regarded as a cluster beginning and center. Each subsequent sample is compared with this cluster and integrated (cluster centroid value is updated) if its distance from the cluster center is below a defined threshold (known as the deviation threshold).

Note: the greater the amount of data included in the cluster, the "heavier" it becomes and the less likely its centroid is to "move" with the addition of new samples.

If the cluster represents a duration greater than the "minimum duration to start a fixation", it is deemed to represent the start of a fixation.
A tolerance criterion allows you to accept samples above the deviation threshold, providing the following samples are below it (external samples are then integrated into the cluster).
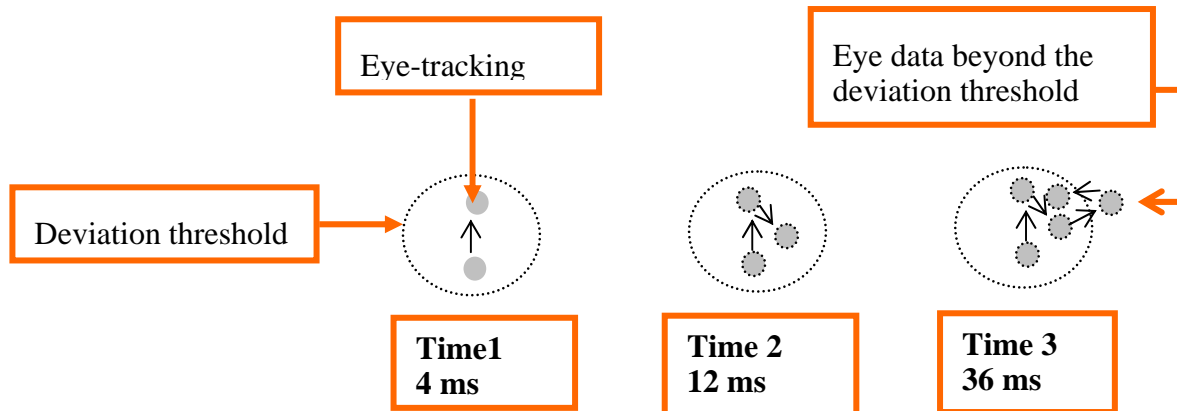
Figure 70: Steps for building a fixation.

With a deviation threshold of 250 (tablet units), a minimum duration of 8 milliseconds and a noise tolerance of 4 milliseconds, we can state that:

- at Time 1, no fixation exists: data duration in the "diameter" threshold duration is below 8 milliseconds;
- at Time 2, with new data, the start of a fixation is "detected";
- at Time 3: the fixation "grows" with the addition of two more items of data, including one external item because the following item is located within the "packet" in less than 4 milliseconds.

To build fixations, click on the "Tools" menu and select "*Build fixations (barycenter method)*". The following dialog box will then be displayed.
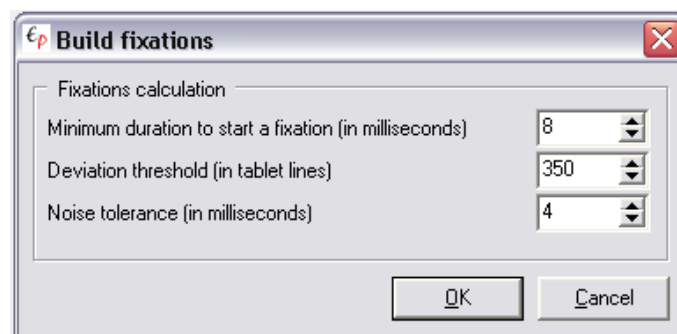
Figure 71: Building fixations using the "barycentric" method.

Fixation building is modulated by three parameters:

- minimum duration for starting a new fixation (in milliseconds);
- deviation threshold (in tablet units) : radius around the center of the fixation;
- Noise tolerance: number of samples "above" the deviation threshold that can be admitted as "members" of the current fixation if subsequent samples fit inside the cluster.

How do I define a deviation threshold? A practical example.

I recorded a participant's writing on an LCD tablet (Wacom Cintiq18SX) and eye movements with an S.R. Research Eyelink2 eye tracker.
I wanted my deviation threshold to be at a viewing angle of 2°.
To be able to calculate what this angle represents on the tablet's screen, I needed some physical measurements. Thus, I measured the distance between the participant's eye and the center of the tablet screen (30.4 cm), then the display width (only the "active display part", not the whole screen "box"): 33.5 cm.

The remaining information can be found in the *Protocol | Description* menu, under the *Specifications* tab.

It says that the eye tracker has ranges of 1024*768, the tablet has ranges of 7220*5784 (resolution is 200 lines/cm) and its display has ranges of 1024*768.

When these data are fed into the *? | Conversions | angle -> Pixel* tool, it tells us that 2 degrees correspond to 34 pixels. But how many tablet lines do 34 pixels represent ?

I wrote 34 in the bottom right-hand box ("pixels" column) in the *? | Conversions | tablet units <-> pixels* menu and clicked on the "<-" button and up came the result: 240 lines.

VII.2. Multi-threshold method

(*Tools/Build fixations (multi-thresholds method)*)

This method relies on the same general principles as the "Barycenter" method, but is more "conservative": it uses a double threshold principle that allows you to integrate a few "out-of-fixation" samples into the calculations.

Method:

To begin a fixation clustering process, there must be more than **n** samples (expressed as a duration in the "*Minimum duration to start a fixation*" field) in an area determined by sample dispersion (deviation) around the barycenter ("*Max initial Standard Deviation threshold*").
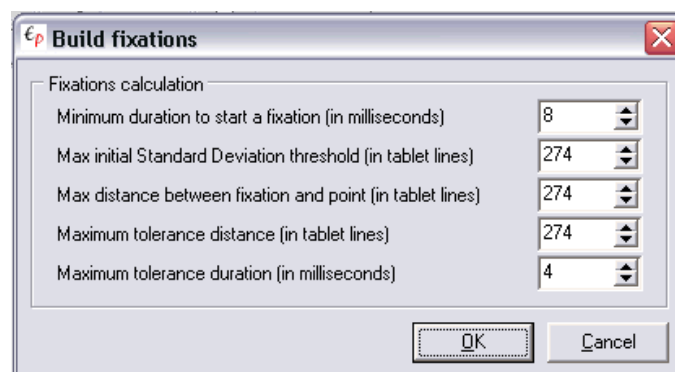
Figure 72: Building fixations using the "multi-threshold" method.

The distance of the following samples from the fixation barycenter is computed to see whether they should be included in the fixation's final calculation or not.

A tolerance criterion ("M*aximum tolerance distance*") allows you to accept samples whose distance is below this criterion but above the threshold ("M*aximum distance between fixation and point*") if the gaze position returns to the "acceptable distance" zone within a predetermined length of time ("M*aximum tolerance duration*"). If it does not, the fixation is ended. The barycenter of the "out-of-zone" group of samples is then computed. If the distance between the latter and the fixation barycenter is below criterion 2 (maximum tolerance distance), these samples are included in the final computation.

Finally, the barycenter of the fixation is calculated.

VIII. BUILD DOWN PAUSES
(*Tools/Build down pauses* menu)

Building a "pen down" basically consists in grouping data that we regard as representing the same event: the pen stops moving at a given place for a given duration.

This tool aims to compensate for the excessive sensitivity of some tablet models. For example, if a participant is asked to keep the pen still on the tablet in a vertical position, the coordinates may not be stable and instead vary around a central ("middle") position, due to slight vertical tremor on the part of the participant. However, as far as the study of writing is concerned, the coordinates of the pen tip on the surface should not change.

Method

This method belongs to the same family as the fixation computing process (cf. "Barycenter method", p. 128): if several successive points are located within a defined perimeter,they are deemed to be part of a single "pen down pause". The whole group of points is then substituted by its barycenter.

Caution: the warning about building fixations also applies here, because building down pauses modifies the number of samples/events in the protocol.

Before using this tool, you should be aware that there are two inherent difficulties:

1) because of the type of method used, the duration of an existing "pen down" pause may be lengthened by the addition of the duration of subsequent points that fall below the deviation threshold;

2) slow tracing may be discretized, i.e. cut into a series of successive pauses. In which case it is advisable to choose as small a "D*eviation threshold*" as possible and a sufficiently large value for "*Minimum duration to start a pause*". You may find help in choosing this value in the relevant literature. For example, when studying the handwriting of adults, a pause lasting more than 150 milliseconds can be regarded as relevant.
This tool should therefore be used carefully and only when needed.

To build pen down pauses, click on the "*Tools*" menu, then on "*Build down pauses*". The dialog box (below) will appear:
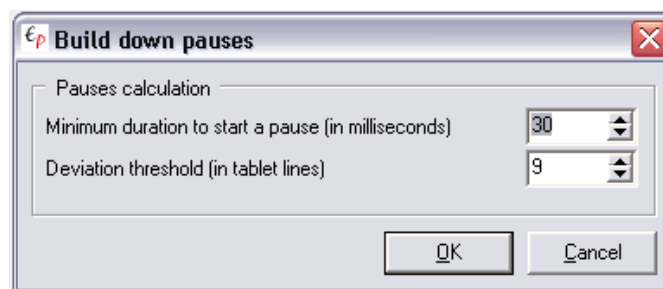
Figure 73: Building down pauses.

"Pen down" pause computation is based on two parameters:
- The minimum duration for starting a pause (in milliseconds).
- A deviation threshold (in tablet units): a radius around the first point regarded as the pause origin. To help you determine the deviation threshold, you can use the tablet test menu (see Tablet visual test, p. 35). For example, given that the smallest distance reported by my tablet is 8 units, I would choose 8, with the aim of calculating pauses taking data clustered within a "1 point" radius into account.

IX. PREVIEW

IX.1. Introduction

The Eye and Pen preview functions allow you to represent data in a static and visual way, in three modes:

- As <u>circle</u> projections: fixations are represented as circles with a diameter reflecting the duration (optional). Eye movements can be traced from one fixation to another.
- As a <u>temperature</u> "map" showing the importance of events in relation to the total length of the protocol. The darker the color, the shorter the duration, the redder the color, the longer the duration.
- As <u>graphs</u> showing the gaze and pen position movement over time.

Note: the Preview mode can be used in conjunction with the "*Shift layers*" tool (see p. 136)
The screen display is composed of the background picture (or background color) and the participant's final product.

To make this easier to grasp, all the following illustrations are taken from the same protocol.

IX.2. Preview configuration

See <u>chapter 1</u> p. 51.

IX.3. Preview in "Circle" mode

(*Protocol/Preview/Circle* menu*)*

You can launch this preview to show:
- pauses;
- fixations ;
- AOI scanpath;

IX.3.1. Pauses

(*Protocol / Preview / Circle/Pauses* menu**)**

Pauses durations are represented as circles with their diameters indexed on the pauses durations (the longer the pause is, the bigger the circle is).
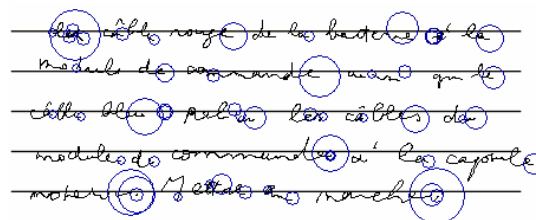
Figure 74: Pause preview with empty circles.

IX.3.2. Fixations

(*Protocol / Preview / Circle/Fixations* menu**)**

Given the options chosen, you may get something like this:
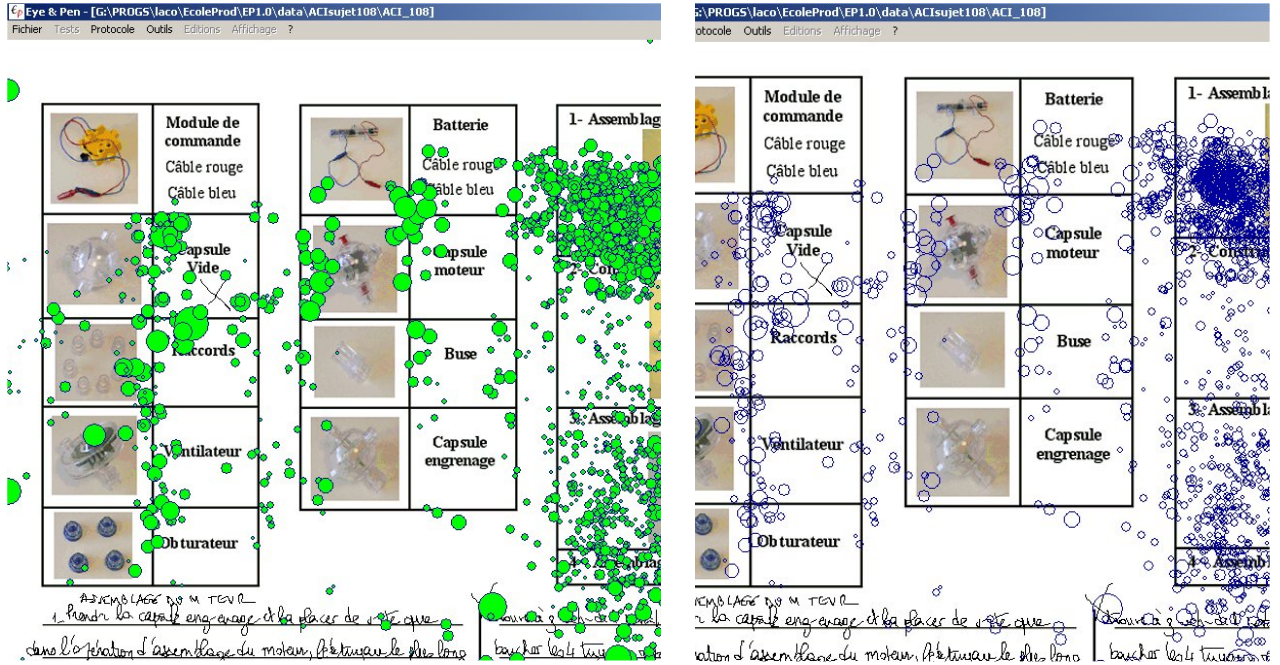
Figure 75: Fixation preview.

If you select the option to include eye movements, you may get something like this:
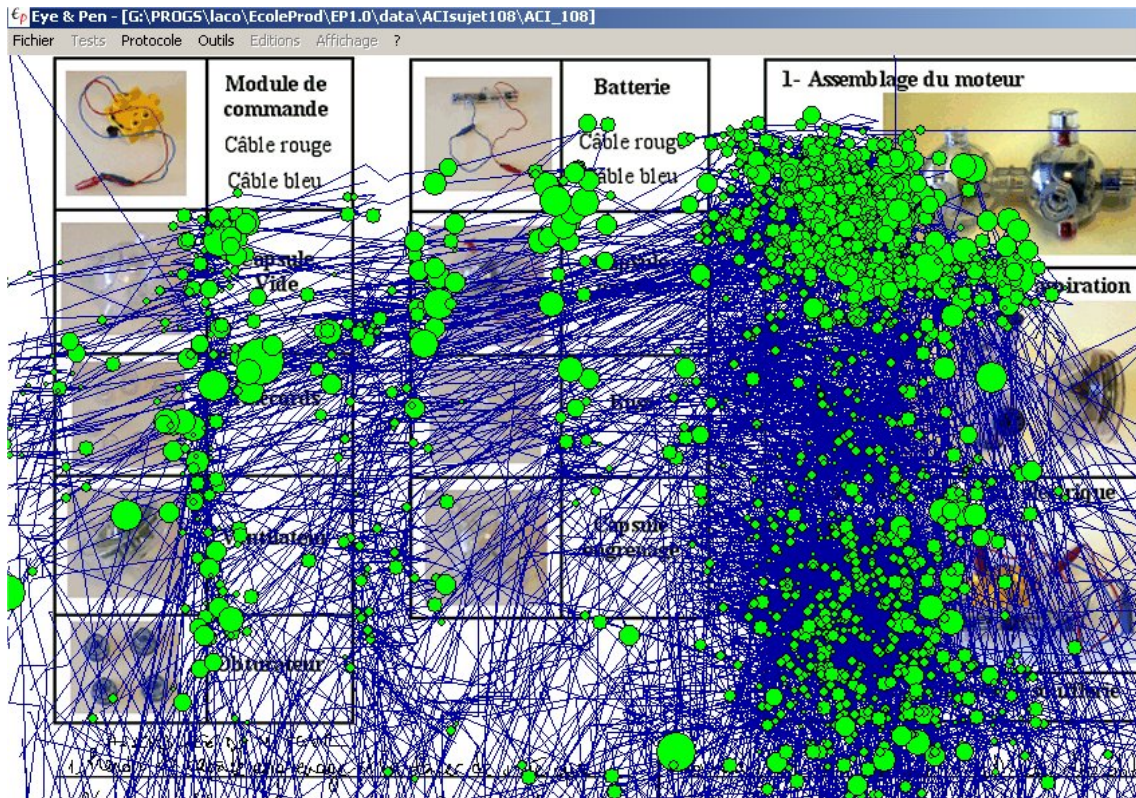
Figure 76: Preview of fixations and eye movements.

IX.3.1. AOI scanpath

(*Protocol / Preview / Circle / AOI scanpath* menu**)**

To obtain a useful and meaningful preview, you must first build fixations and define the AOIs.

You will then be able to visualize groups of fixations and their route across the AOIs. Each series of fixations on the same AOI is combined into a group, represented by a circle located at the coordinates of the center of that group.

Depending on the chosen options (see Preview configuration, p. 51), the size of the circle representing a group of fixations will be relative to the summed duration of the fixations in the group. Successive groups will be connected by a line and the circles will be numbered in chronological order.
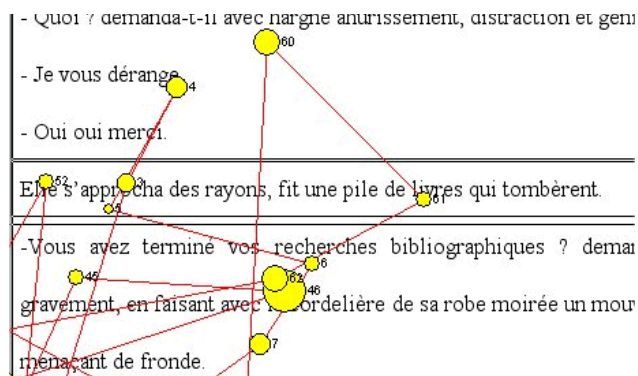
Figure 77: Preview of AOI scanpath.

IX.4. Preview in "Temperature" mode

(*Protocol/Preview/Temperature* menu*)*

In this preview mode, durations are represented by colors, as a "thermograph".
You can launch the "Temperature" preview showing:
- Pauses
- Pauses classified according to duration
- Fixations (takes the foveal vision size into account)

IX.4.1. Pauses

(*Protocol/Preview/Temperature/Pauses* menu)

Color represents pause duration in relation to the whole set of pauses. Pauses are represented by a rectangle at the place where the pen left the tablet or became immobilized.
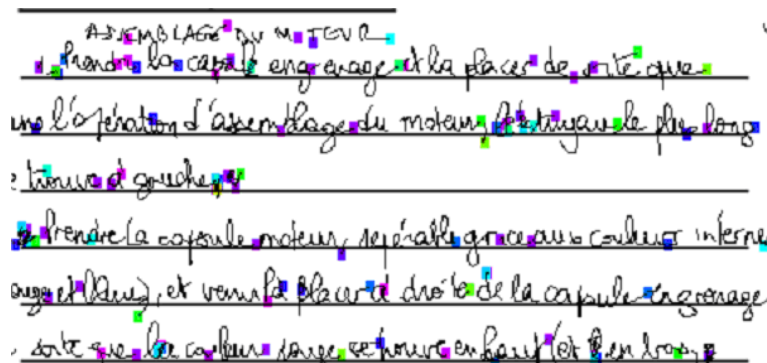
Figure 78: Pause preview.

IX.4.2. Pause duration categories

In this preview mode, pause durations (in milliseconds) are classified as follows:

The right section of the scale shows the color for pauses lasting 2,000 milliseconds or more.
Below is an example of how they are represented:
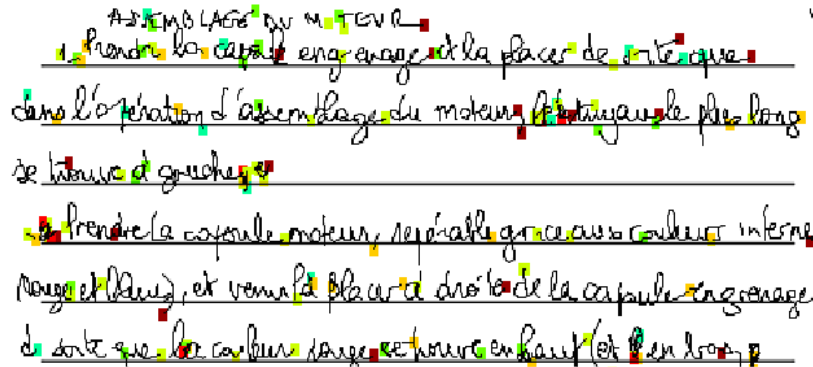
Figure 79: Preview of pauses, color-coded according to duration.

IX.4.3. Fixations

This preview mode only shows the fixation durations on the protocol.
Below is an example of a protocol preview in this mode (foveal vision diameter is 30 pixels; see).

Figure 80: Fixation preview in "temperature" mode.

"Hotspots" represent the areas of greatest fixation duration.

IX.5. Data preview as "Graph"

(*Protocol /Preview / Graph* menu)

Previews available in the submenus of the Preview/graph menu are all curve-based representations.

Graph axis scales are automatically resized according to the data being displayed.

Be careful about memory capacity: displaying a graph with 6 curves for a 517,000 data protocol (total of Tab and Eye data for a full page of writing produced in about 20 minutes) takes up approximately 130 Mb of RAM memory.

Graph windows may be resized with the mouse: when the mouse cursor hovers over the lower right-hand corner of the windows, it turns into a diagonal double arrow. Press the left mouse button and, keeping the button pressed, move the mouse in the direction you wish the windows to resize. The window will resize in line with your movement.

Inside the windows, you can zoom in/out and have the graph scroll up/down and horizontally, as is the case for the graphs displayed for the protocol statistics (see p. 152).

IX.5.1. XY coordinates

(*Protocol / Preview / Graph /XY Coordinates* menu**)**

This visualization shows how the pen and gaze coordinates change over time in the X and Y dimensions.
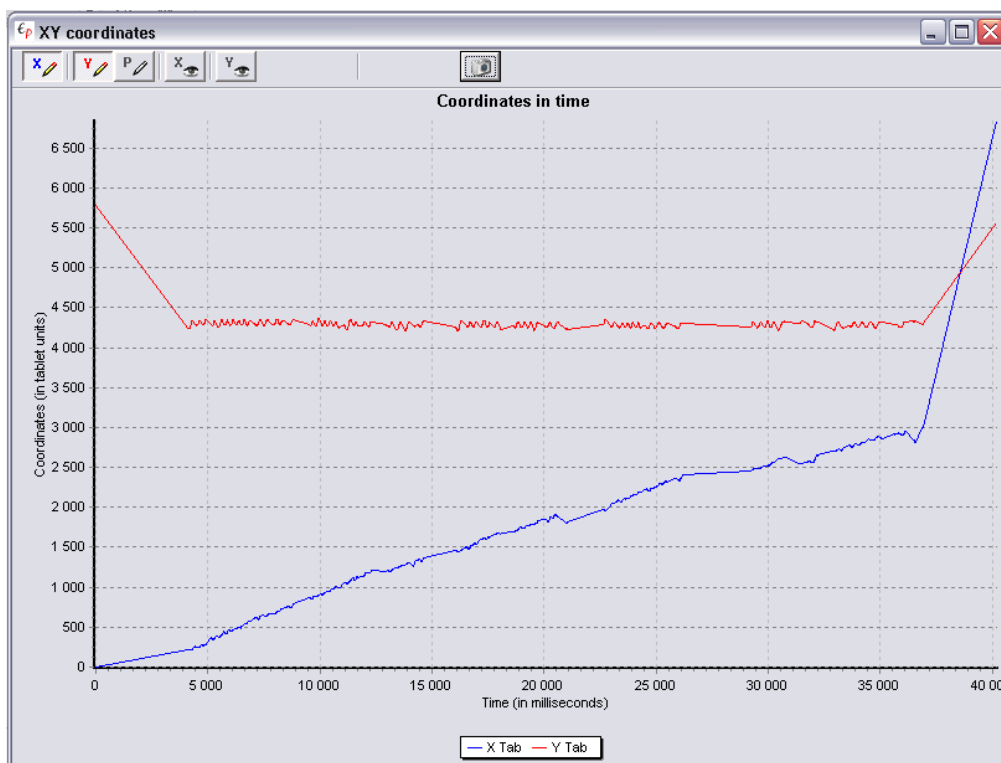
Figure 81: XY coordinates graph.

The buttons on the toolbar located at the top of the graph allow you to select the curves to be displayed.

BUTTON	DESCRIPTION
[X icon]	Horizontal movement of the pen (X). This curve is selected by default.
[Y icon]	Vertical movement of the pen (Y). This curve is selected by default.
[P icon]	Pressure exerted on the pen tip.
[X eye icon]	Horizontal gaze movement.
[Y eye icon]	Vertical gaze movement.
[camera icon]	Capture the current graph and save in a JPG picture file. A dialog box allows you to choose the name of the file and the folder in which to save it. The protocol's name is proposed as a default name.

IX.5.2.   Gaze-pen distance

(*Protocol / Preview / Graph / Gaze-Pen distance* menu**)**

This graph displays the distance between gaze and pen over time (data can be filtered with a weighted moving average, i.e. the equivalent of a 1st order low pass filter).
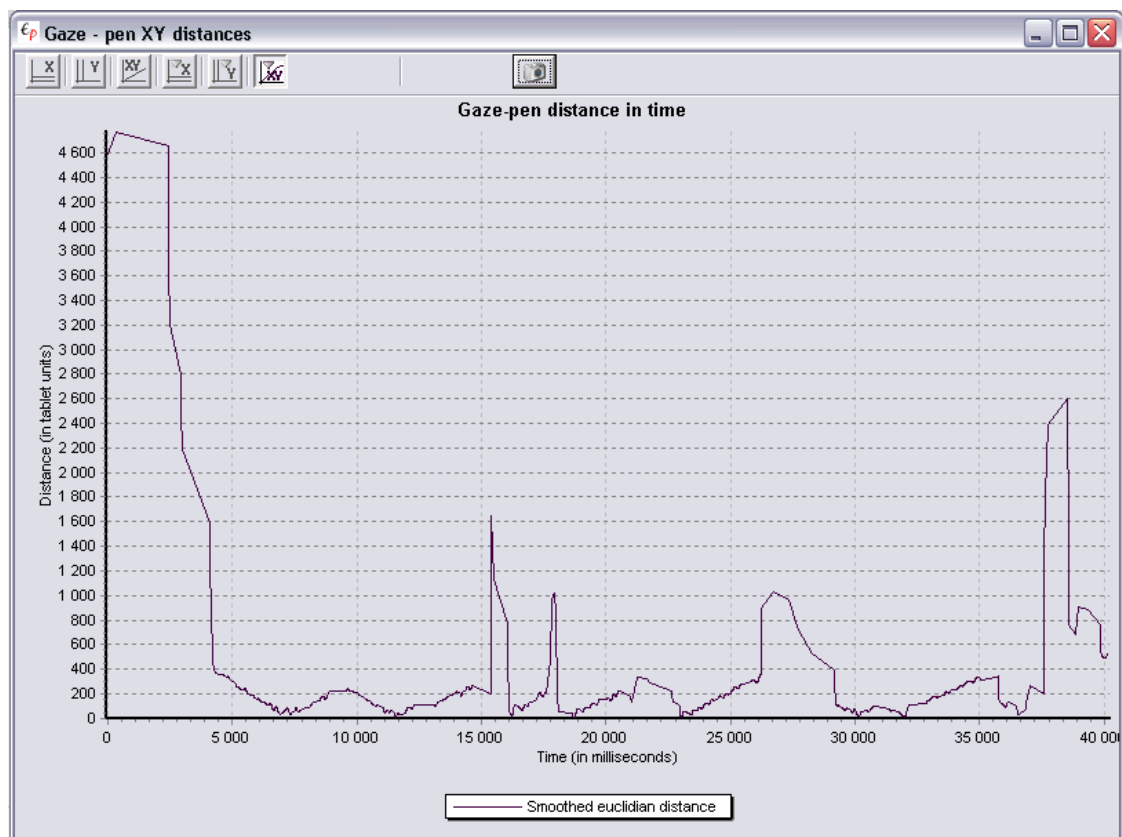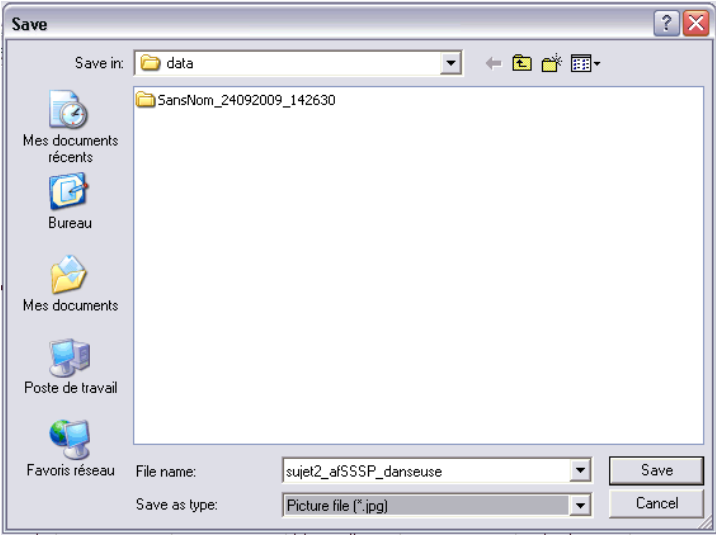
Figure 82: Gaze-to-pen distance graph.

The buttons on the toolbar located at the top of the graph allow you to select the curves to be displayed.

BUTTON	DESCRIPTION
⊥X	Changes in the horizontal distance between gaze and pen tip positions as a function of time.
‖Y	Changes in the vertical distance between gaze and pen tip positions as a function of time.
XY	Changes in the Euclidian distance between gaze and pen tip positions as a function of time.
⊥X	Changes in the horizontal distance between gaze and pen tip positions as a function of time. The curve is smoothed with a weighted moving average.
‖Y	Changes in the vertical distance between gaze and pen tip positions as a function of time. The curve is smoothed with a weighted moving average.
XY	Changes in the Euclidian distance between gaze and pen tip positions as a function of time. The curve is smoothed with a weighted moving average. This is the default display.
📷	Capture the current graph and save in a JPG picture file. A dialog box allows you to choose the name of the file and the folder in which to save it. The protocol's name is proposed as a default name. *Figure 83: Saving the graph as a JPG picture file.*

X.   SHIFT LAYERS

(*Tools/Shift layers* menu)

During data acquisition, the data may not necessarily be where they should be (they have "slipped") because:
- the sheet of paper on the tablet has moved (or been moved);
- even if the participant's head has been immobilized, s/he has nevertheless succeeded in moving slightly;
- etc.

Eye-tracking data therefore move in relation to the tablet data and background picture.

If the participant was writing on a sheet of lined paper and we use an image of this sheet as a background for analysis (see [configuration](#)), we can see that the writing is not where it should be.

To make it easier to track down these "displacements", apply the following tip prior to data acquisition:

Put (draw) a cross in the middle (or a more convenient place) of the production medium – sheet of paper, background picture, etc. - and ask the participant to fixate this mark for approximately 20 seconds. Next, ask the participant to draw over this cross with the pen. You can now start your experimental recording.
The participant's first fixation will therefore be on the cross. If a shift of data does occur, this first fixation will be a good way of spotting the problem and helping to align the data. The principle is the same for both writing and drawing: the participant's cross must be on top of your cross!
After the experimental recording, ask the participant to draw the cross again and to fixate it. This will be a way of checking that the data are still "well in place".
To limit this risk, you can try "fixing" the paper, e.g. using plastic corners on the tablet into which you can slide the sheets.

In the analysis stage, a "Temperature" preview (see page 132) may help you detect this kind of flaw and correct it by shifting the layers.

Recall: to have all the data in the same coordinates system, eye tracking data must be converted into the tablet coordinates system.

We regard the protocol as being made up of 3 superimposed layers: the screen background (color or background picture) represents the bottom layer, the tablet data are "sandwiched" in the middle layer, and the eye tracking data make up the top layer (third level). The reference level is the tablet data layer.

To re-adjust these layers, click on the "*Tools*" menu, then select "*Shift layers*".
The following dialog box will be displayed.
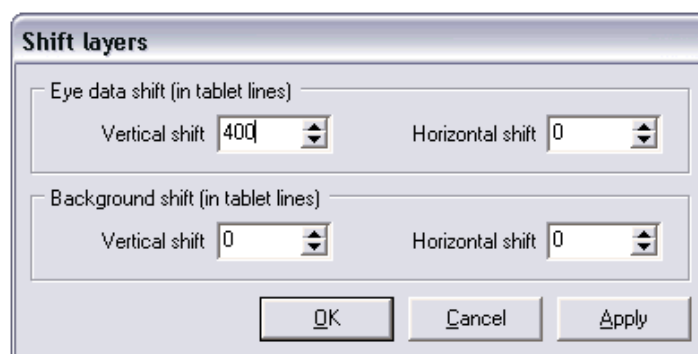
Figure 84: Dialog box for adjusting the position of the layers.

This box has two frames (parts), labeled:
- Eye data shift (in tablet lines): horizontal and vertical movement applied to all eye samples.
- Background shift (in tablet lines): horizontal and vertical movement applied to the background picture.

As the tablet data layer is the reference, it is never "moved".

There are two ways of shifting the eye data or the background layer: using the computer mouse or using numerical values.

Numerical shift of the layers

Set the number of tablet units you want the layers to be moved by, in the "horizontal shift" and/or "vertical shift" boxes.

Click on the "*Apply*" button to... apply the desired shift to layer(s).

Caution: the "*Apply*" button will validate your entries, but there is then is no way back except by setting new values.

If the applied shift has the desired effect, click on the "*OK*" button to close the dialog box. If not, modify the values again or click on "*Cancel*" to close the dialog box.

Visual (and convenient) layer shift

Click with the right mouse button on the "start location" of the shift.
Then, keeping the button pressed down, move the mouse to the place where you want that point to be located. A dotted line will connect the starting point to the mouse cursor.
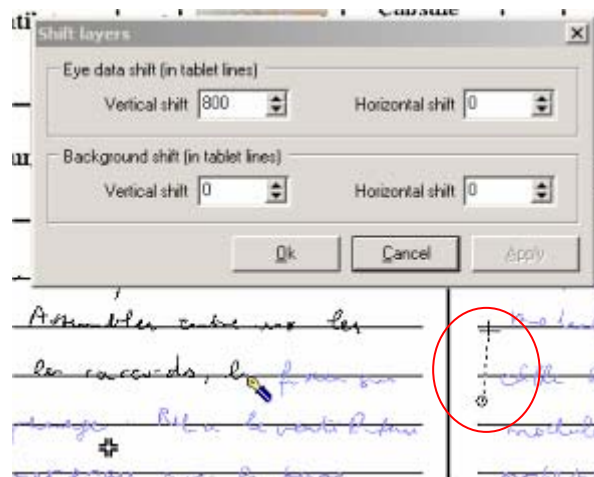
Figure 85: Setting a layer shift using right mouse button selection.

Release the mouse button and go the contextual menu (pops up to the right of the mouse cursor) if you want to shift the eye-tracking data layer or the background picture.
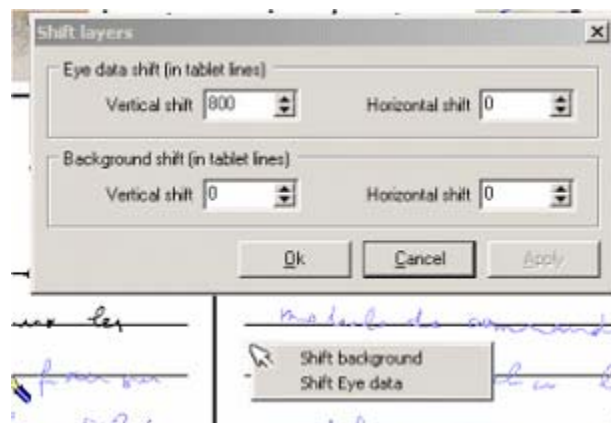
Figure 86: Selecting the layer you want to move using a contextual menu.

Values in the "*Shift layers*" dialog box are automatically updated and applied.

XI. DATA CODING

XI.1. Manually-coded data

The coding tool provides information on the event currently being observed in the protocol, but also allows you to attach a code to the data.
Below the "*D*" box of each data type (Tab or Eye), you will find a column with a list of numbers (from -1 to 127).
This column is used to assign codes to the data.

Each data item can be given a code from -1 to 127.

By default, each data item is coded "0" (neutral code).

Every data item coded -1 will be excluded from calculations and editing.

To code a data, click on a number in the column corresponding to its type (Tab or Eye).

Each item can only have one code.

XI.2. Coding an item of tablet data when in trigger zone 1 or 2

(*Tools/Tablet zone auto-coding* menu)

The "Eye and Pen" program has tools which automatically code the tablet data "passing into" trigger zones 1 or 2 (defined in the acquisition configuration panel, "Simple" tab).

Click on the "*Tools*" menu, then select *"Tablet zones auto-coding..."*.
The following dialog box will be displayed.

Figure 87: Automatic coding of data in trigger zones.

Enter a code number (between 1 and 127) in the "*Pen down in trigger zone 1*" box.

Enter a code number (between 1 and 127) in the "*Pen down in trigger zone 2*" box.

Click on the "*OK*" button to automatically attach these codes to the tablet data "located" in these zones.

Note:

Each line of the dialog box is enabled only (and thus the corresponding zone taken into account in analyses) if the corresponding "trigger zone" option is selected (ticked) in the Simple paradigm configuration (*File/Configuration/Acquisition/Simple* menu*)*.

XI.3. Assigning the −1 code to "Eye" data beyond the calibrated area

(*Tools/Code -1 "out-of-field" Eye data* menu)

If you want to automatically assign the code **-1** to eye-tracking data located beyond the calibrated area (i.e. generally speaking, outside the analysis screen), there are two solutions:

- tick the "*Code -1 out-of-field data"* box in the "*New Analysis"* tab of the analysis configuration panel (optionally tick "*aggregate successive out-of-field data*");
- click on the "*Tools*" menu, then on "*Code -1 out-of-field Eye data*".

The of using the "*New Analysis*" configuration for automatic coding is that you have to be sure that the data are properly "set up" (see "Shift layers", p. 136), otherwise you risk tagging valid data as invalid.

Please note that the "-1" code can be assigned to any type of data.

XI.4. Coding "Eye" data in the Visual Zones (AOI)

(*Tools/Visual zones (AOI)* menu)

A Visual Zone (often found as "*Area of Interest*" in the literature, or AOI) is a rectangular area of a protocol in which we consider the eye movements and fixations to be of particular interest.

To automatically code Eye data in a particular zone of the protocol, you must first define this Area Of Interest with the help of a specific tool. All data in this zone will then be coded with the associated code.

Important:
AOI-based processing is managed according to the order of AOIs in the list. When two zones share a common area(s), the last AOI in the list will "have the last word" when it comes to coding.
e.g. if, after a first zone has been defined, a second zone is defined, partially overlapping zone 1, data in this "dual zone" area will be given zone 2 coding.

To define an AOI, click on "*Tools*" and select "*Visual Zones (AOI)*".
The following dialog box will be displayed.
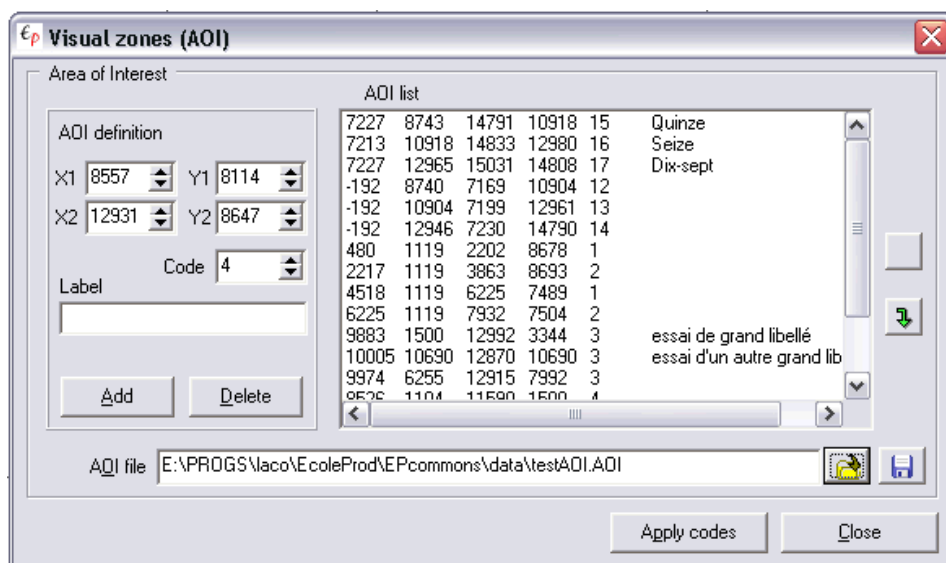
Figure 88: AOI definition/selection panel.

To define an AOI, follow the steps below.

STEP	DESCRIPTION
1	Click with the left mouse button on the place where you want the AOI to begin (its upper left-hand corner) in the protocol.
2	Keeping the button pressed down, move the mouse toward the right, then down. You have now created a rectangle which you can adjust by moving the mouse.
3	When you are satisfied with the result, release the mouse button. The coordinates of the selected zone will be updated in the boxes labeled "*X1*", "*Y1*", "*X2*" and "*Y2*".
4	Type the code value you want to be associated with this zone in the "*Code*" box.
5	Type a label (optional) in the "label" field. This label is only used to help manage AOI easily and more "readable" than numbers. It is not used in data extractions.
6	Click on the "*Add*" button to add this new AOI to the list of AOIs (in the frame labeled "*AOI list*"). Each line of this list contains "*X1*", "*X2*", "*Y1*", "*Y2*" and "*Code*".

You can save or load your AOI lists in order to use them again another time.

To save an AOI list, click on the "floppy" icon in the lower right-hand corner of the dialog box. A "*Save as*" dialog box will be displayed. Choose a name for the file and click on "*Save*". The full path and filename of this file will be updated to the right of the "*AOI file*" label.

To load an AOI list you created earlier, click on the "opened folder" icon. A dialog box will be displayed where you can select an "AOI" list file.
Click on the "OK" button to load and open this list. The "*AOI list*" will be updated.

Hint: the keyboard shortcut of the label "A<u>O</u>I file" (<alt+o>) select the "Open" icon. Then, you just have to pres on the <enter> key to open the dialog box to load an AOI list file.

To select an AOI, click on the corresponding AOI line in the "*AOI list*". This AOI will be displayed on the protocol as a "darkened" rectangle.
The Up and Down arrow to the right of the list allows you to move up or down the items in the list. This means you can modify the order of items in the list.

To delete an AOI, select it in the "*AOI list*", then click on the "*Delete*" button in the "*AOI definition*" frame.
The selected AOI will be erased from the list.

If you click on the "*Close*" button without having clicked on "*Apply codes*" before, the AOI list won't be lost. If you re-open the panel, you'll find it again.

When defining AOIs, make sure you take into account the participant's foveal vision radius. An over-restrictive definition may lead to under-estimated gaze durations in these zones.

For example, I want to study gaze durations for a photo of a desk. I am particularly interested in an object lying on the desk. I therefore define an AOI around this object. When the participant's gaze approaches the "boundary" of this object, the coordinates of the gaze may be a few pixels outside the AOI, even though the participant is actually looking at this object. The amount of data "involved" will therefore be under-estimated. The reason for this is that the participant not only sees what is at the exact center of his/her line of sight, but also what is within a certain perimeter, i.e. foveal vision.
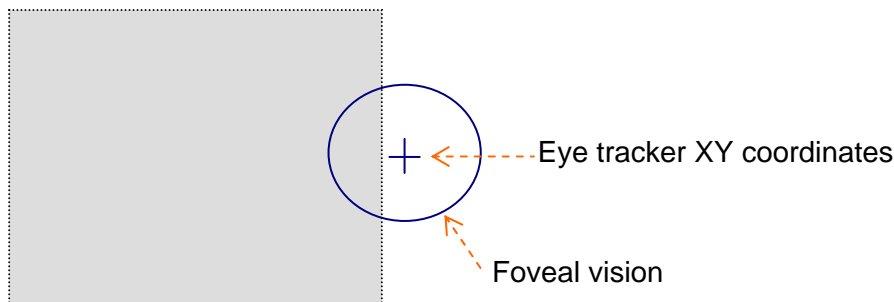
Figure 89: Foveal vision at the edge of a visual zone (AOI).

You may therefore need to define an AOI that is larger than the item you wish to study.

AOI and background picture:

if you save an AOI list in the directory where the background picture is, using the same name, you can use the option *"If an AOI file exists for the current background picture, apply to data"* (*configuration/Analysis/New analysis*, see page 45).

XI.5. Erasing all the codes

(Tools/Erase codes menu)
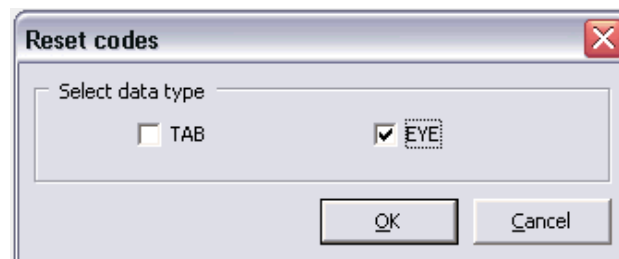
Figure 90: Selecting the type of data for resetting the codes.

Tick the data category (Tab and/or Eye) where you want the data codes to be **reset to zero**. Simply click on the "OK" button: now, none of the data in the selected data type has a significant code.

XI.6. Replacing one code with another

(Tools/Replace codes menu*)*

You want to substitute one code for another for a particular type of data (Tab and/or Eye). The following dialog box will be displayed.
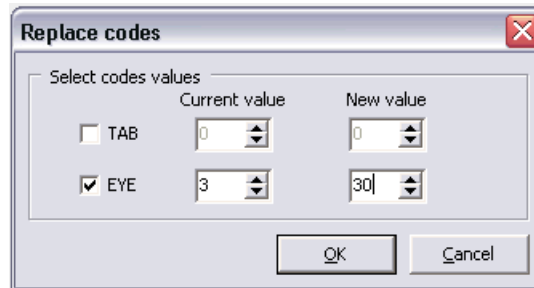
Figure 91: Replacing the codes given to the data types.

To change a code, follow the steps below.

STEP	DESCRIPTION
1	Choose the data type you want the modification to apply to, by ticking the "*Eye*" and/or "*Tab*" boxes.
2	In the "*Current value*" column, enter the code value(s) you want to be replaced.
3	In the "*New value*" column, enter the new code value(s).
4	Click on the "*OK*" button to apply the modification to your data.

XI.7. Word separation

(Tools / Word separation Menu)*

The word separation tool is <u>not</u> an optical character recognition (OCR) tool.
It is a semi-automated method of "locating" words.
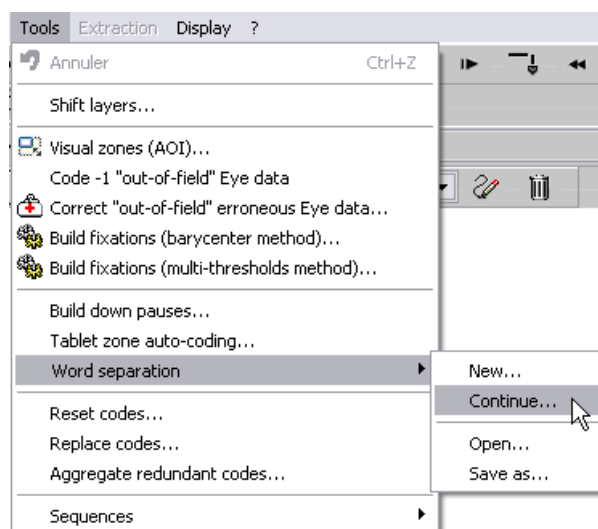
Figure 92: Word separation management menu.

To handle the word separation data, you can use the "*Tools/Word separation*" menu that allows you to start/resume loading or save these data.

Based on this "segmentation", you will have the option of:
- creating sequences
- coding Tab data at word boundaries
- coding the pauses between two successive words
- building AOIs around these words.

The "*save as*" submenu allows you to save the current word separation data in a different (separate) file. Type a name for the file and click on "Save" to validate.

The "*Open*" menu allows you to load a previously saved word separation data file.
A dialog box appears in which you can select a word separation data file (*.WDS). Click on "Open" to validate and the data will be loaded.

Word separation data files allow you to have more than one set of separation data for a single protocol, which you can then use according to your needs (e.g. different types of data extraction). You can even use the same separation data for different protocols (e.g. if a participant wrote word in "fields").

Separation data are automatically saved (and reloaded) with the protocol (as for sequences).

Word separation is carried out in two steps: the first step consists in overall, automated location (the best adjustment that correctly separates the most words), while the second step consists in manually refining the separation and adding information for each individual word.
The first step is managed via the "*New*" submenu, whereas the second step is initiated via the "*Continue*" submenu.

XI.7.1. Step 1: overall tuning

Word separation basically consists in locating the basic units made up of writing data between two "pen up" pauses.
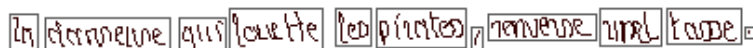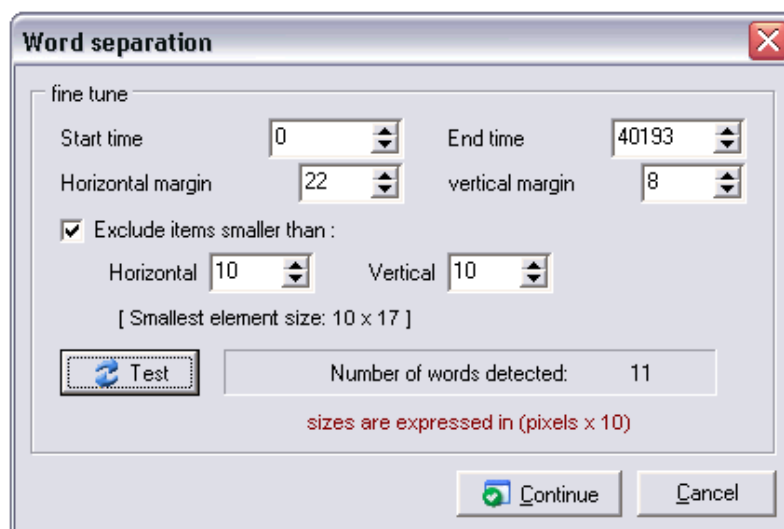
Figure 93: Word separation, step 1.

This first dialog box is used to manage the overall word scanning. As soon as it appears, an initial separation is proposed (in the protocol, words are surrounded by a border to visualize the result of the separation process, as in the above figure).

If the proposed separation satisfies your needs, click on the "*Continue*" button. If not, you can modify and edit the separation parameters and click on the "test" button to see the result of the separation with your new parameters.

The "*Fine tune*" panel fields and buttons are explained below:

LABEL	DESCRIPTION
Start time	By default, this field indicates the protocol's start timestamp. You can edit this field's content to restrict the separation to a certain period of time within the protocol.
End time	By default, this field indicates the protocol's finish timestamp. You can edit this field's content to restrict the separation to a certain period of time within the protocol.
Horizontal margin	Defines the horizontal space added to the size of each word. If the edges of two "words" touch and are temporally successive, they are grouped together as a single word. This parameter helps, for example, to "stick" together parts of a word that have been separated (writing style), etc.
Vertical margin	Defines the vertical space added to the size of each word. If the edges of two "words" touch and are temporally successive, they are grouped together as a single word. This parameter helps, for example, to "stick" together signs added to the nearest words, such as dots on "i"s, etc.
Exclude items smaller than	If you tick this checkbox, all the "words" that are smaller than the horizontal and vertical minimum sizes that you have defined will not be isolated as "words". This function may come in useful for discarding punctuation signs, for example.
[Smallest element size]	Shows the horizontal and vertical sizes of the smallest "word" in the current separation configuration. This information may come in useful for adjusting edges or defining the size of elements to be excluded.
Test	Click on this button if you have modified any of the parameters in the "Fine tune" panel, in order to update the display of the word separation result.

Once the parameters have been adjusted, click on the "Continue" button.

A new window will then enable you to fine-tune (or correct) each separated word (you can directly access this step via the "*Continue*" submenu, if the first step was completed earlier).

XI.7.2. Step 2: word-by-word adjustment

Fine-tuning, word by word, allows you to modify the events regarded as beginnings and endings, codes and attached labels. "Up" pauses are used as delimiters, but are not included within the word boundaries.
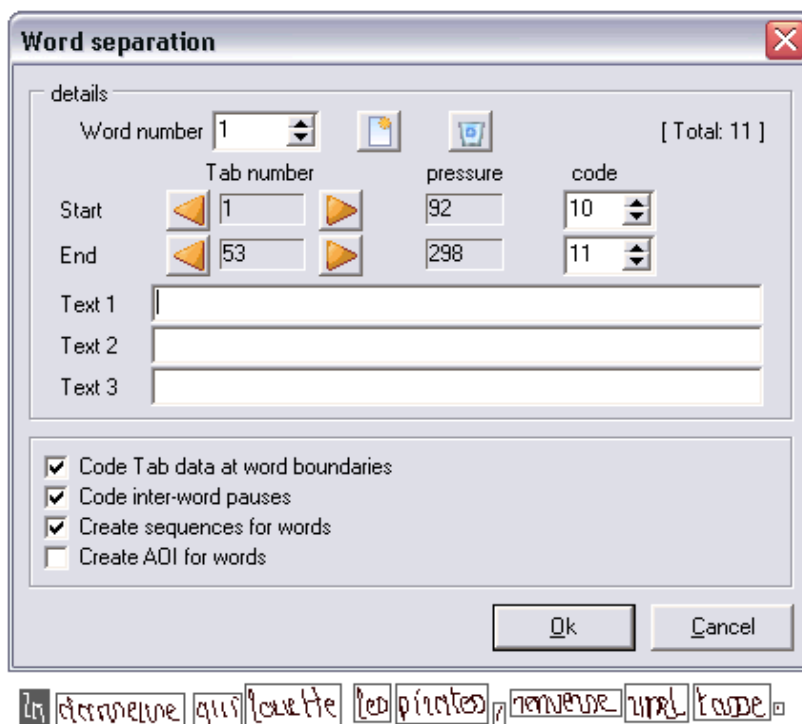
Figure 94: Word separation, step 2.

This panel helps you to review each word and add one or more labels to each one.

The fields and buttons in the "Details" panel are described in the following table:

LABEL	DESCRIPTION
Word number 1	Shows the number of the current "word". Click on the Up and Down arrow on the right side of the field to move forward/backward from word to word. You can also use keyboard shortcuts: ○ Alt + left arrow: previous word; ○ Alt + right arrow: next word; ○ Alt+End: last word; ○ Alt+Home: first word. When the current word changes, the cursor moves to the "Text 1" field.
	Clicking on this button will create a new word. It will be inserted after the current word (highlighted). All subsequent words will be re-numbered from the current word onward (you will find a more detailed explanation below).
	Clicking on this button deletes the current word. All subsequent words will be re-numbered, the word next to the current word taking its number and so on (you will find a more detailed explanation below).

[Total]	Shows the number of separated words.
Start	Shows data relating to the beginning of the current word.
End	Shows data relating to the end of the current word.
Tab number ◀ ▶	Tablet data number in the protocol. Buttons to the left and right of the number allow you to move forward or backward through the tablet data (where possible).
Pressure	Shows the value of the pressure exerted on the pen for this tablet data.
Code 10 ⬍	This field displays the code value that will be associated with the tablet data if you select the "Code Tab data at word boundaries" option (first and last tablet data of the word). To change this value, click on the Up or Down arrow button to the right of the field or type a value of between 0 and 127.
Text	Descriptor (label) linked with the word. The number of lines available to describe the "word" is defined in the *Configuration / Analysis / Tools* menu. If you wish to generate sequences based on "word separation", these lines will be assembled and given a single sequence descriptor.

XI.7.2.1   delete a word

To delete a word, make it the current word. In the following example, we wish to delete the coma (word number 7). We use the arrows ⬍ in the "Word number" field to select this word.
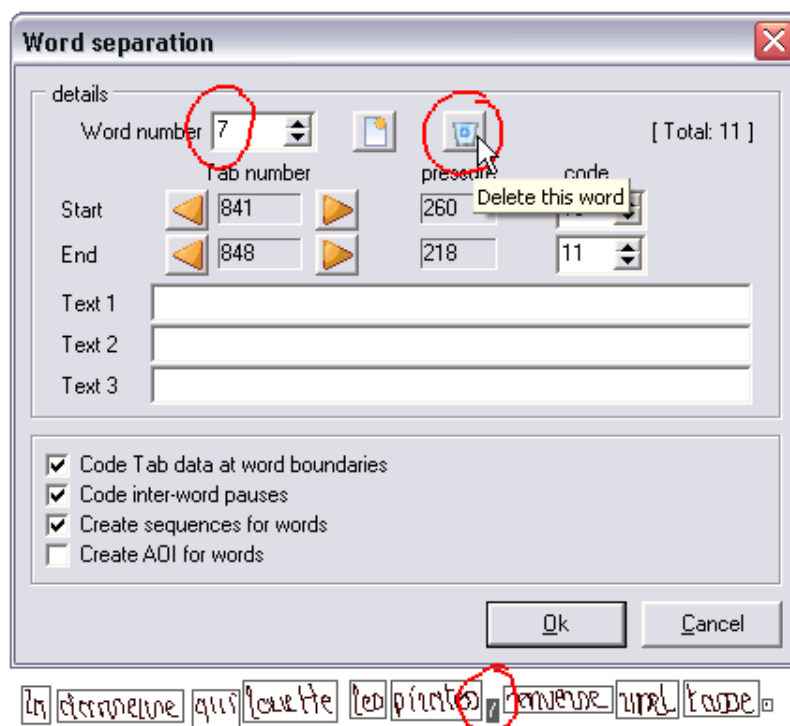
Figure 95a: A "word" deletion.

Next, we click on the "trashcan" icon to delete the word. The word that was previously word no.8 becomes the new number 7. Each of the subsequent words will be re-numbered as well.
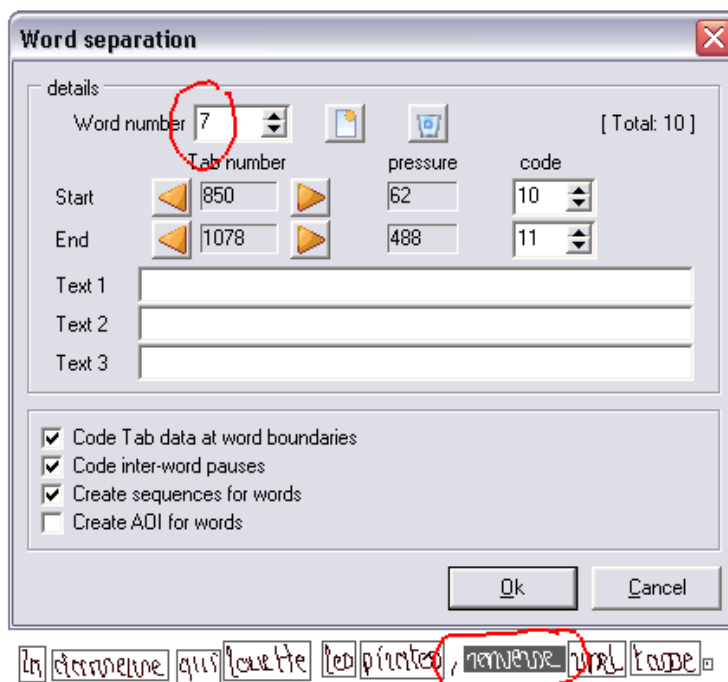
Figure 95b: Deletion of a "word" (new current word).

XI.7.2.2 Inserting a word

To be able to insert a word, there must be enough room, otherwise room must be made for it.

In the following example, the "overall" separation (step 1) combines two written words within a single "unit" (highlighted): word 5, "les pirates" ("the pirates", in French).
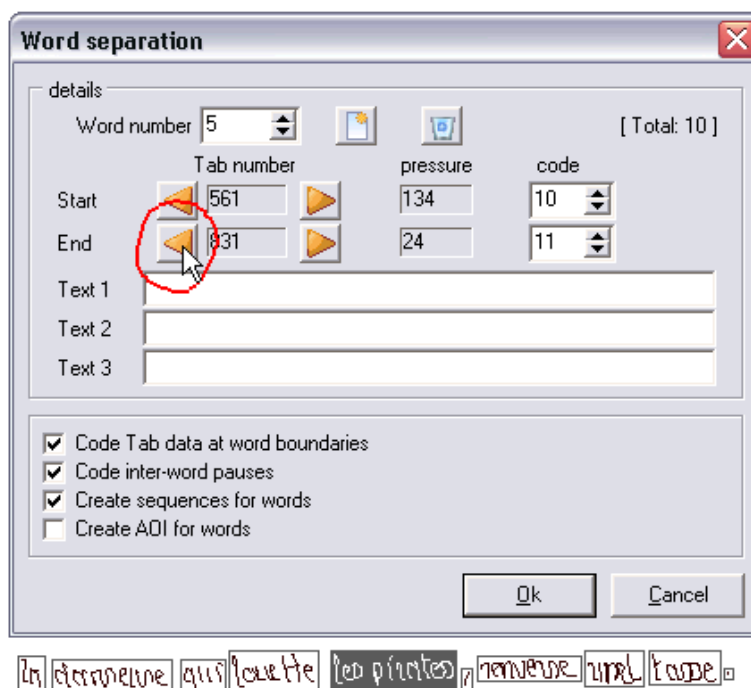
Figure 96a: Inserting a "word" (resizing).

We can "shrink" this unit to the first written word with the help of the button of the end of word Tab data. By clicking on this button, we move the end of word to the left (the selection width is reduced). In the process, we also free up some space in the protocol (no longer assigned to "separated words").
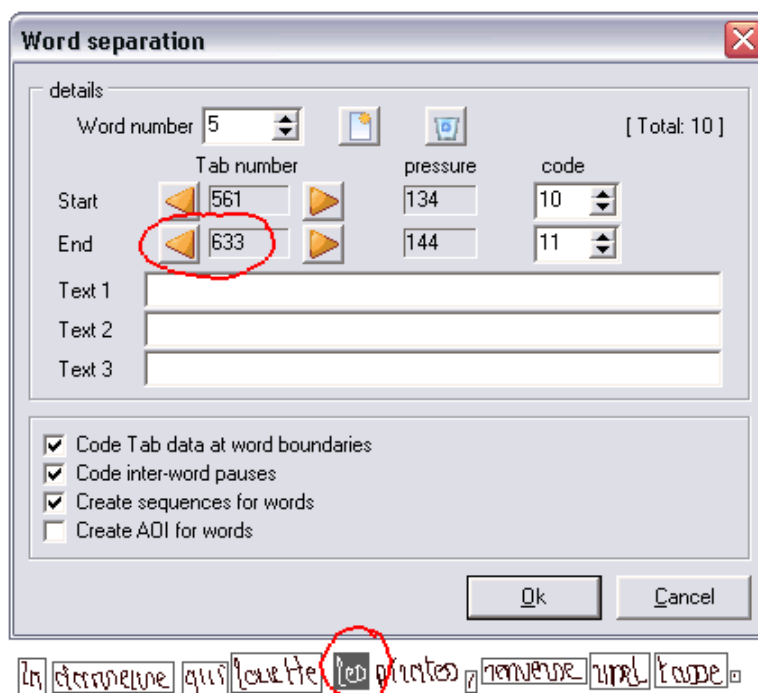
Figure 96b: Inserting a "word" (size of word reduced).

Insertion can now take place in the space freed up after the curent word. We click on the button to insert a new word. The new word's Tab data will encompass the whole space left free between the current word and the next word.
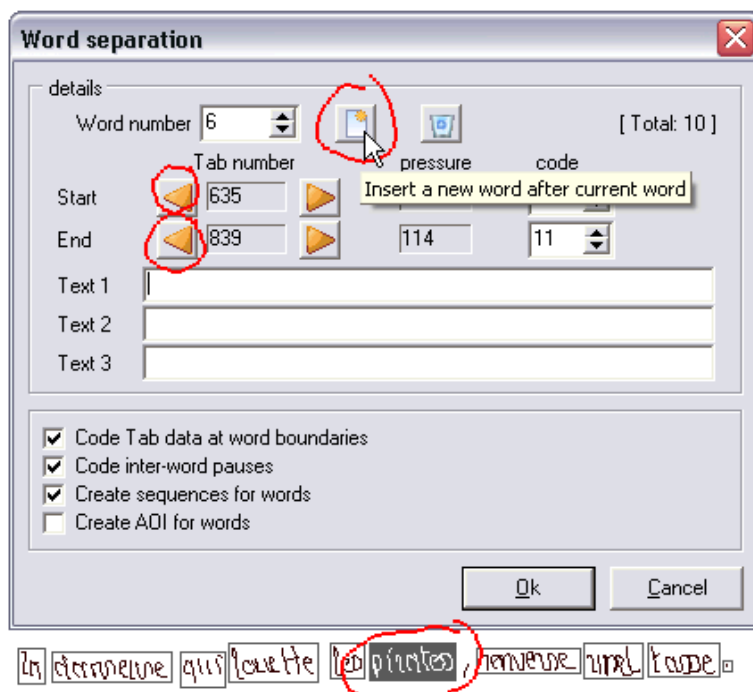
Figure 96c: Inserting a "word" (new word is added).

The inserted word becomes the new no. 6 and each of the following words is re-numbered.

XI.7.2.3   What to do with separation data

The frame at the bottom of the window lists actions that can be performed based on word separation data when the "OK" button is clicked:

LABEL	DESCRIPTION
Code Tab data at word boundaries	If this checkbox is ticked, the codes associated with the first and last Tab data for each word will be applied to the protocol coding. These codes are visible in the coding tool (p. 116) when you "navigate" in the protocol.
Code inter-word pauses	If this checkbox is ticked, the pauses (Up or Down) between two "words" will have their codes in the protocol replaced with inter-word codes (defined in the *File/Configuration/Analysis* menu, *Tools* tab).
Create sequences for words	If this checkbox is ticked, a sequence will be generated for each "word" and its descriptor will be appended to the list of sequences. The start time of the sequence will be the timestamp for the word's first Tab data (or of the pause preceding the word, depending on the chosen options). The sequence's end time will be the timestamp for the word's last Tab data.
Create AOI for words	If this checkbox is ticked, a visual zone (AOI) will be generated for each word and its descriptor will be appended to the protocol's list of AOIs. The AOI coordinates will be the coordinates of the frames (borders) around each word.

If no action has been selected when you click the "OK" button, a dialog box will prompt for a confirmation before closing.

For example, you may not want to create sequences or codes for data, but simply record the separation data for a colleague who will fill the text fields in later, prior to generating sequences.

The *File/Configuration/Analysis* menu, *Tools* tab, "word separation" frame (see p. 53) allows you to determine the options relating to this section.

XII. STATISTICAL PROTOCOL DESCRIPTION
(Protocol/Description menu*)*

In order to help you analyze tablet and eye-tracking data, Eye and Pen can provide statistical descriptions of the data.

Note: events coded -1 (Tab and Eye) are excluded from descriptions.

The following description panel will be displayed.
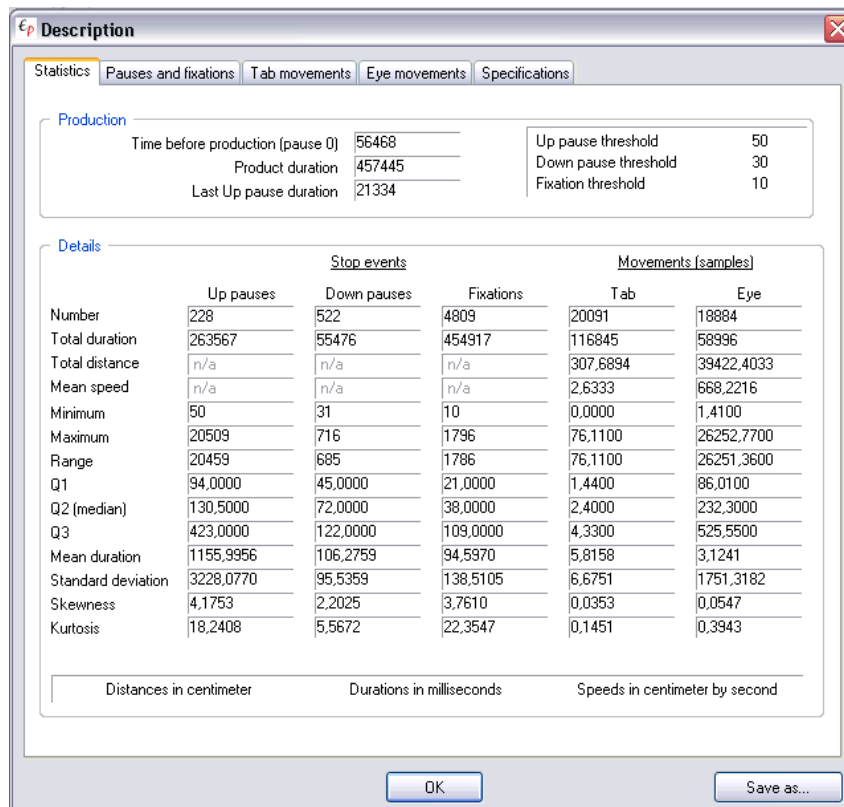
Figure 97: Statistical description.

The first tab (Statistics) shows quantitative data in two frames:

- "*Production*": general information about the entire protocol;
- *"Details"*: statistical description of the tablet and eye-tracking data;

The "*Production*" frame contains three types of information:

- pause "zero" duration, also called pre-writing pause (time elapsed before the participant presses the pen on the tablet);
- production duration (pause 0 excluded);
- last pause "Up" duration of the production. This pause may not be the last event immediately preceding the recording stop if the recording was stopped by an "Escape" key press or if the participant slipped the pen into the "end" zone without lifting it up.

Hint: place a thick border (e.g. made of plastic) around the "end" zone to force the participant to raise the pen in order to press it in this "box".

The "*Details*" frame gives statistical parameters for different categories of information:

- the "stop" events during the written production (Up or Down pauses) and ocular activity (fixations). These parameters are filtered by (see Thresholds);
- the movements of the "eye" or pen.

For each of these categories, the following information is given:

- Number: number of events in the category;
- Total duration;
- Total distance (for "movements");
- Mean speed (for "movements");
- Minimum, maximum, range;
- Quartiles: Q1, Q2 (median) and Q3, the values dividing the distribution into 4 groups. These values are computed on durations for "no movement" events and on speeds for "movement" events;
- Mean duration (for "stops");
- Standard deviation;
- Skew: distribution asymmetry indication. A positive value shows an asymmetry with a "swelling" on the left side. The distribution is skewed to the right;
- Kurtosis: indicates the flatness or otherwise of the distribution shape. A high positive value indicates a "pointed" shape with long "tails", whereas a negative value indicates a "square" shape;

The "*Save as*" button allows you to export these statistical data into a text file.

<u>Three different tabs</u> can be used to generate a graphic representation of these distributions:

- "Pauses and Fixations": distribution of pause and fixation durations;
- "Tab Movements": pen movement speed distribution;
- "Eye Movements": eye movement speed distribution;

The last tab "Specifications" details the specifications of the recording devices used.
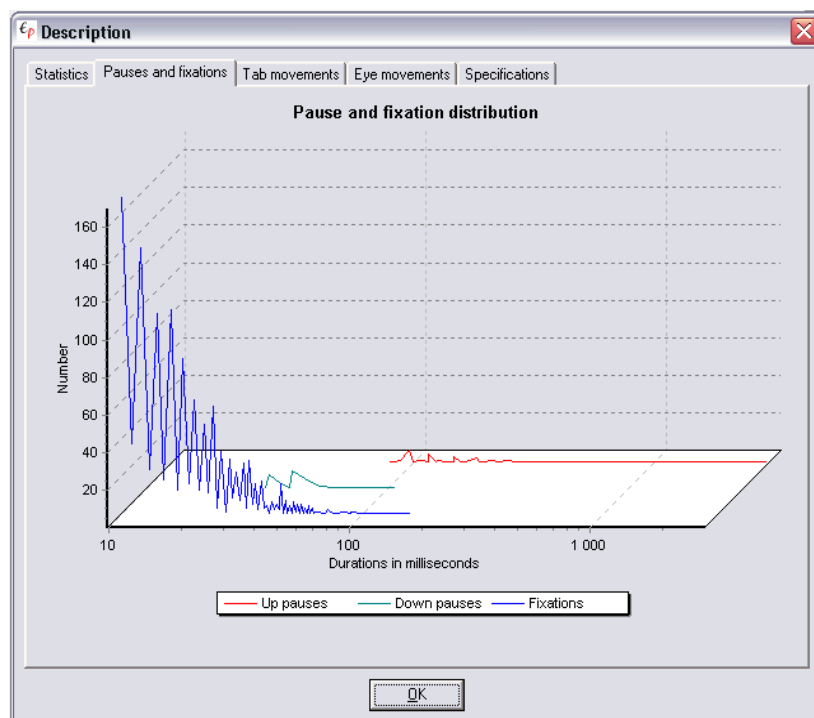
Figure 98: Graph showing pause and fixation distribution (logarithmic scale).

To **zoom** in on the data: using the left mouse button, select the part of the graph you wish to enlarge, moving from its upper left-hand corner to its lower right-hand corner.
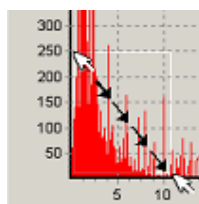
To **cancel**, select part of the enlarged area (its size does not matter) moving from the upper right-hand corner to the lower left-hand corner.
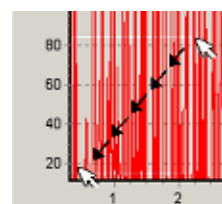
You can move the whole graph with a right mouse click (drag the graph).

"Tab Movements" tab: pen movement speed distribution.
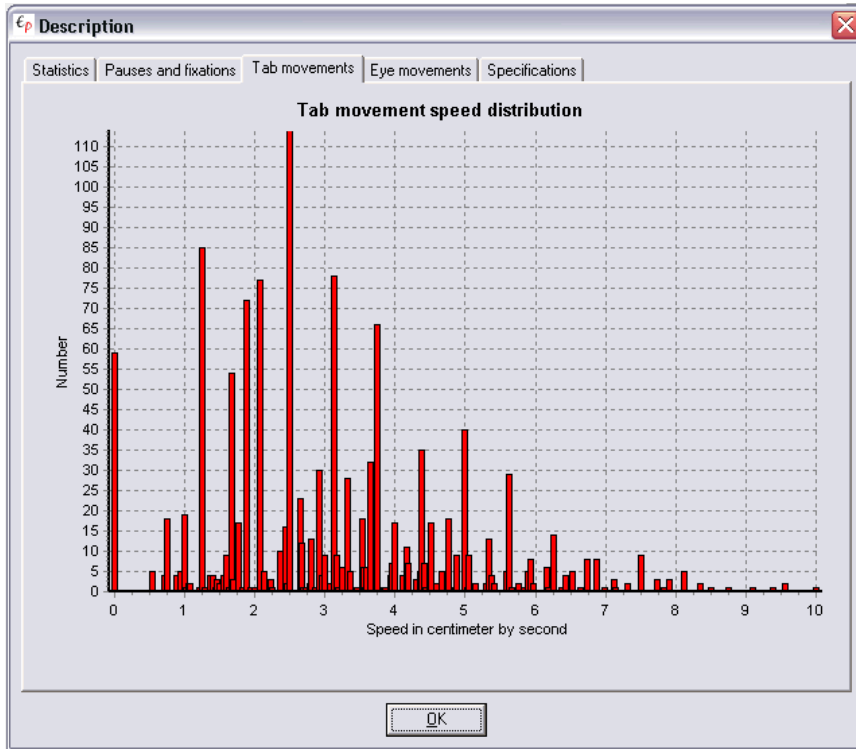
Figure 99: Pen movement speed distribution.

"Eye Movements" tab: eye movement speed distribution.
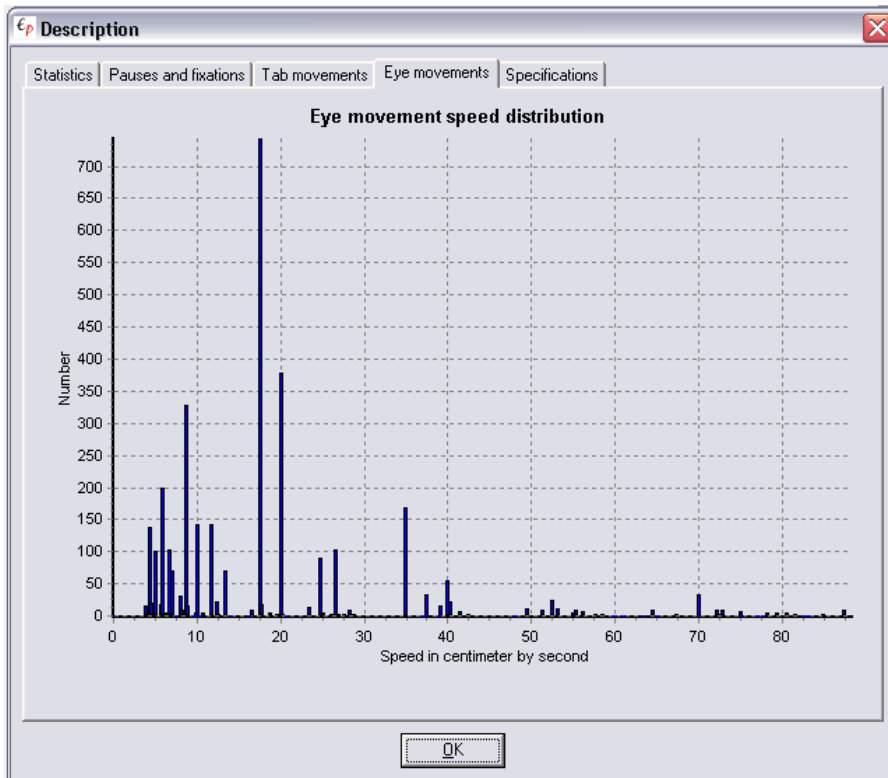
Figure 100: Gaze movement speed distribution.

"Specification" tab: shows the properties of the devices (tablet and possibly eye tracker) used to record the protocol.
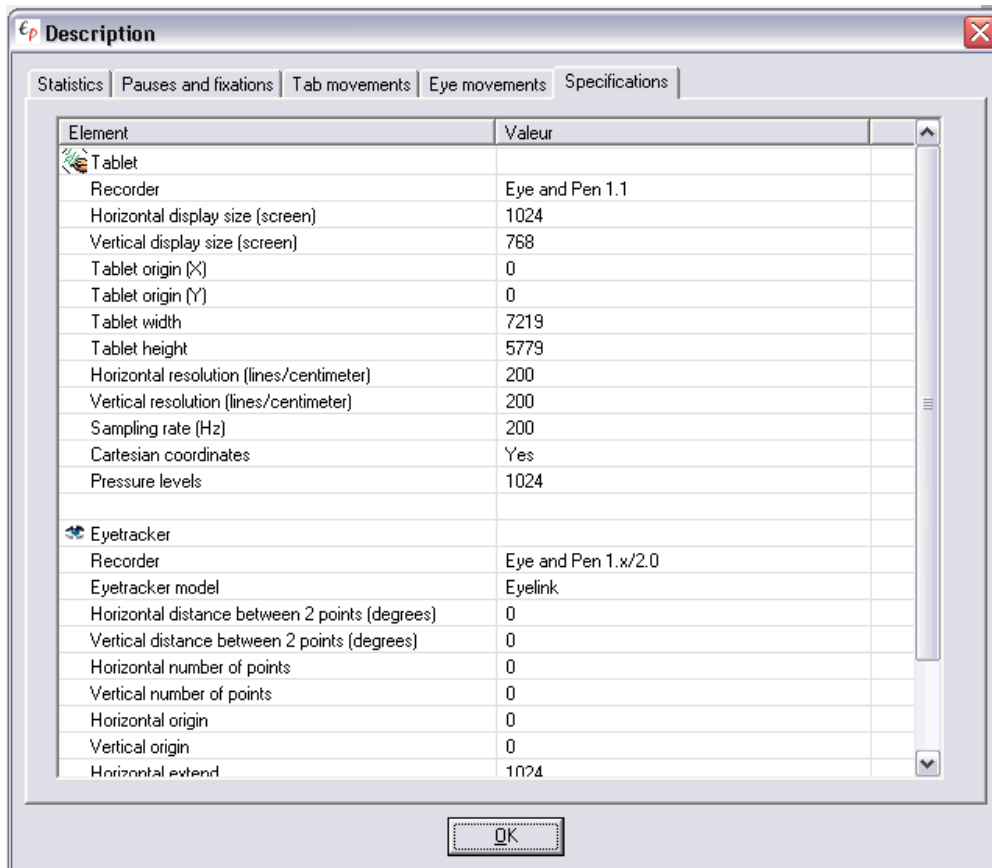
Figure 101: Protocol recording device specifications.

XIII. USEFUL TOOLS AND FUNCTIONS

XIII.1.   Sequences

(*Display/Sequences* menu to show/hide the tool)

Defining sequences is a way of "breaking the protocol down", i.e. creating subsections.

A sequence contains a set of data (Tab and Eye) defined by a beginning time, an ending time and a name (label).

The "*Sequences*" tool allows you to:
- define a sequence;
- directly "jump" to the beginning of a sequence.
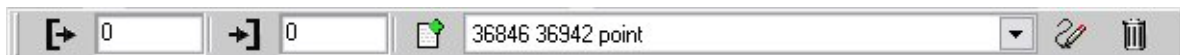
Figure 102: The tool for defining sequences.

ICON	DESCRIPTION
	Set the current time (time elapsed since the start of the data recording, displayed in the "*clock*" tool) in the protocol as the sequence beginning.
	Time at which the sequence begins.
	Set the current time (time elapsed since the start of the data recording, displayed in the "*clock*" tool) in the protocol as the sequence ending.
	Time at which the sequence ends.
	Create a sequence. Once the times of the beginning and ending of the sequence have been defined, you can create a new sequence and give it a name. When you click on the icon, the dialog box below will be displayed: *Figure 103: Describing a sequence.* Type the label for the sequence in the "***Descriptor***" box, then click on the "*OK*" button. The new sequence, with its start and end times and its label, will be added to the drop-down list, to the left of the button

	Edit the selected sequence label. To change the name/label of a sequence, select this sequence from the list (button to view the list) then click on the button**.** The "descriptor" dialog box will be displayed: *Figure 104: Editing a sequence label.* Edit the label for the sequence (the dialog box is horizontally resizable). Click on the "*OK*" button to validate.
	Delete a sequence. Select a sequence from the list (button to view the list). Once selected, click on the trash button.
	Scroll down the sequence list. When you click on this icon, the sequences you have defined will be displayed in the form of a drop-down list. <u>Important</u>: sequences are sorted according to their date of creation (the most recent one will be at the bottom of the list). Click on a sequence. The protocol will immediately jump to the start time of this sequence.

To manage the list of sequences, you may use the "*Tools/Sequences*" menu. This menu has submenus that allows you to open, save, edit and clear a list of sequences.
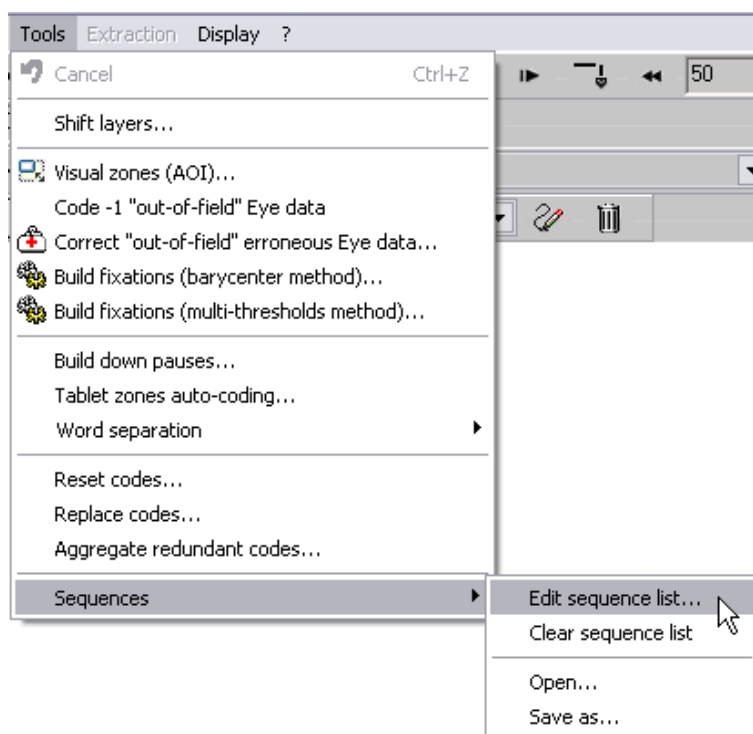
Figure 105: Sequence management menu.

To edit the list of sequences, click on the "*Edit sequences list...*" submenu.
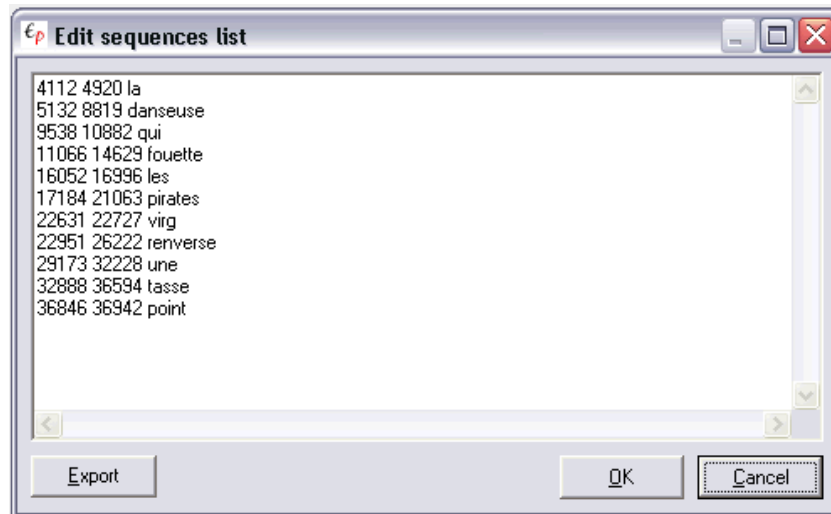
Figure 106: Editing the entire sequence list.

Make sure you respect the order of elements in a sequence: start time (in milliseconds), blank space (keyboard space bar), end time, blank space, and descriptor.

When you have finished editing, click on the "*OK*" button to validate the changes.

The "*Export*" button allows you to save the sequence list in a "text-only" format file.
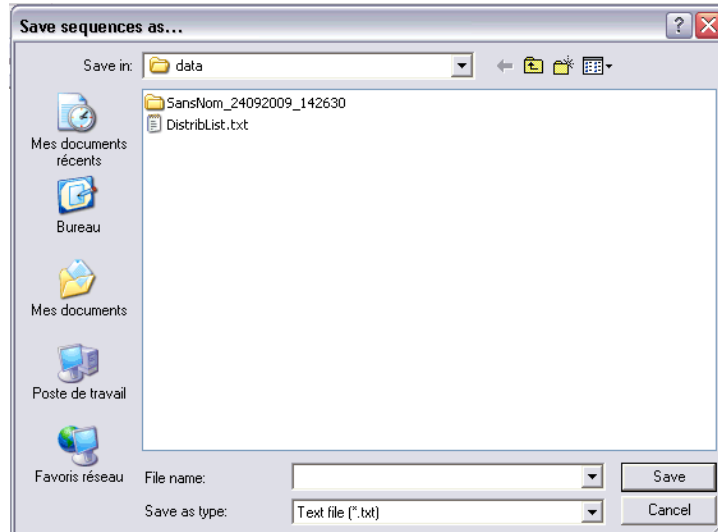
Figure 107: Export sequences as "text only".

You may thus re-use the sequence list data as you want, for example with some other type of software.

To erase the content of the sequence list, click on the "*Clear sequence list*" button.

The "*Save sequences as*" submenu allows you to save the current sequence list in an independent sequence list file.
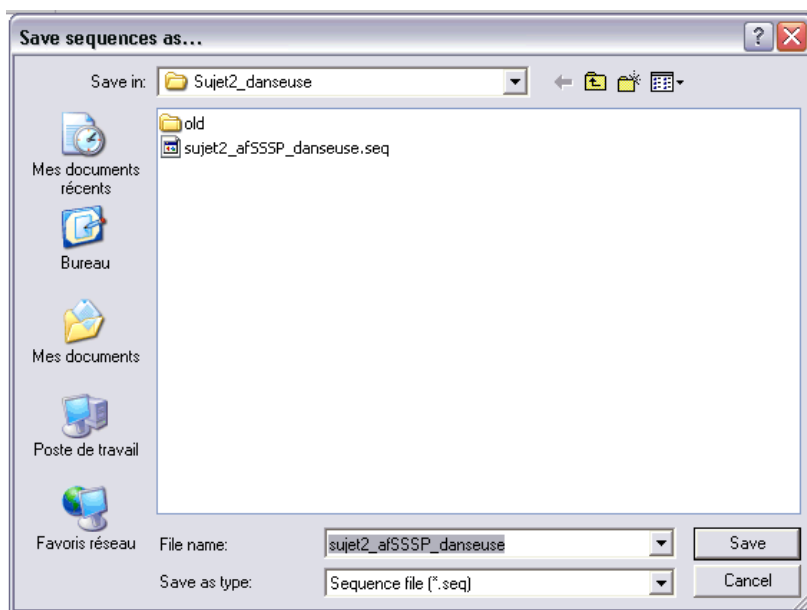
Figure 108: Save sequences in a new file.

Type a name for the file in the "*Filename*" field and click on "*Save*" to validate.

The "Open" submenu allows you to load (or re-load) a sequence list previously saved in a file. A dialog box helps to select such a file (*.SEQ).
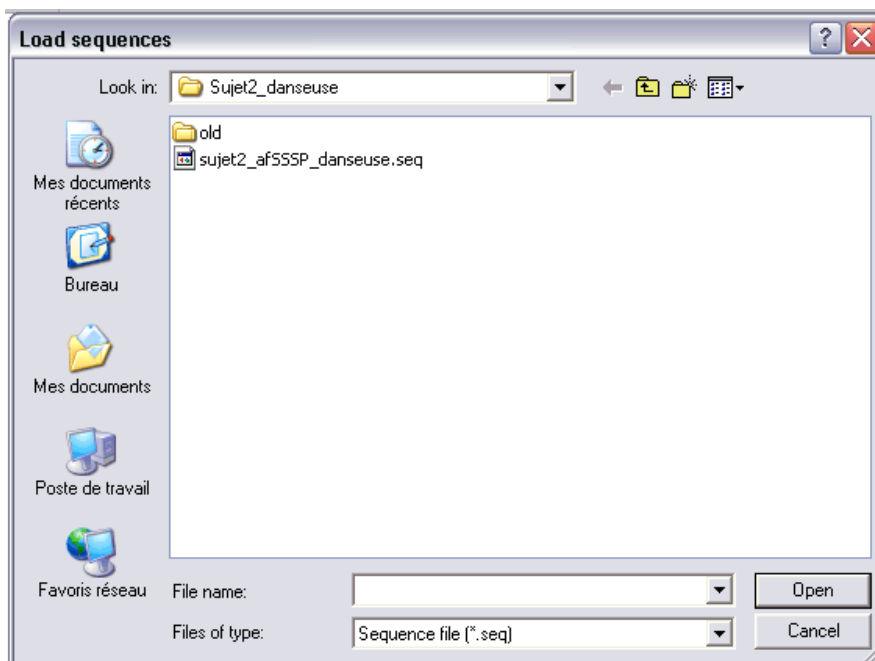
Figure 109: Load a sequence file.

Click on the "*Open*" button to validate and the sequences will be loaded into the sequence list, replacing the existing ones if there was any.

The sequence list files allows you to manage multiple sets of sequences for a same protocol and to use them as a function of what is intended (different data extractions, for example) or to use the same sequences with different protocols.

XII.2. Successive data with the same code aggregation

(*Tools/Aggregate redundant codes* menu)
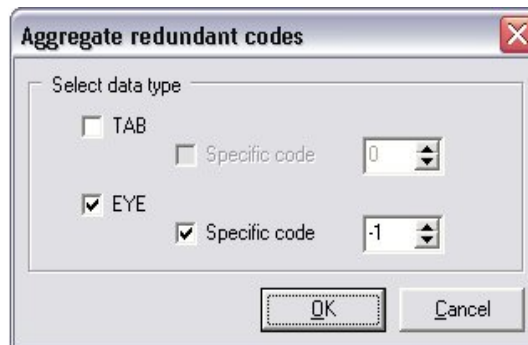
Figure 110: Select code for aggregation.

This function can be used for deleting, for example, successive Eye data with the code -1 (mainly "out-of-field data").
A dialog box allows you to choose the data type it will apply to: Tab and/or Eye.
If no specific code is set ("specific code" checkbox unchecked), treatment will apply to all codes: two successive data sharing the same code will be aggregated.
The aggregation adds the duration of the erased data to that of the first item of data. The data are thus aggregated "to" the first item of data.
This function must be used with care, as it can be applied to all codes (except 0).

XIII.3.　　　　Correct "out-of-field" erroneous Eye data

(*Tools /Correct "out-of-field" erroneous Eye data* menu)

Some eye trackers may return "weird" data when the participant looks outside the calibrated area (e.g. 4068 degrees).
If you do not wish to discard these data (coding them -1), for example to keep the corresponding saccade, you can correct these data by constraining them to a margin around the calibrated area.
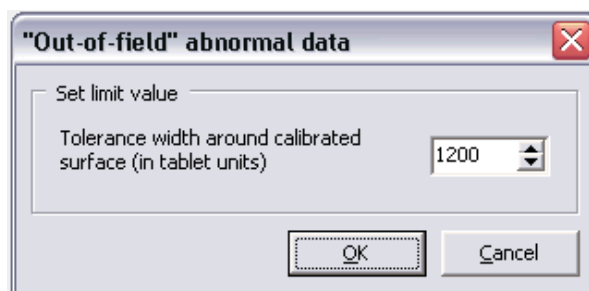
Figure 111: Set the tolerance margin width.

This dialog box allows you to define a tolerance margin around the calibrated area (in tablet units). Data within this margin will not be modified. Only data outside this area will be modified: their coordinates will be replaced using linear interpolation between the last known "good" data (i.e. within the area or margin) and the next good data. If there are no such data (i.e. end of protocol), coordinates will be replaced by the nearest limit of the last "good" data.

XIII.4.   Cancel

(*Tools /Cancel* menu)

As its name says, this function allows canceling an action. The following actions may be undone:

- Shift layers
- Visual Zones (AOI)
- Code -1 "out-of-field" Eye data
- Correct "out-of-field" erroneous Eye data
- Aggregate redundant codes
- Tablet zones auto-coding
- Replace codes
- Reset codes
- Fixations
- Build down pauses
- Thresholds

This function is cumulative (one may undo several successive actions), and the number of undo levels can be set in Eye and Pen configuration (*Configuration/Application* menu).

XIII.5.   Zoom

(*Display/Zoom* menu to show/hide this tool)
This tool allows you to zoom in on a particular area of the screen.
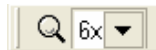
Figure 112: "Zoom" tool.

To select the zoom factor, click on the ▼ icon, then select a zoom factor from the 6 zoom levels (2x, 4x, 6x, 8x 10x, 12x).

Next, click on the magnifying glass to activate the tool (click again to de-activate).
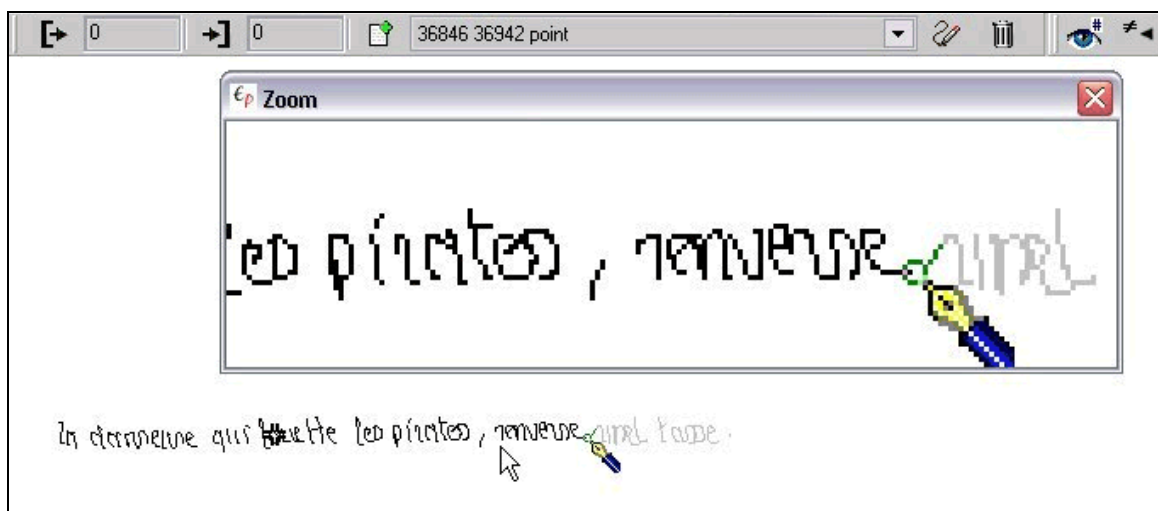
Figure 113: Zoom "at work".

Your mouse cursor will indicate the center of the area to be magnified in the Zoom window.

This window is resizable and movable.

There are two ways of closing it:
- click on the cross in the upper right-hand corner of the window;
- click on the magnifying glass icon again.

XIII.6.　　　　Capture a picture of the analysis

(*Protocol/Capture to picture* menu)
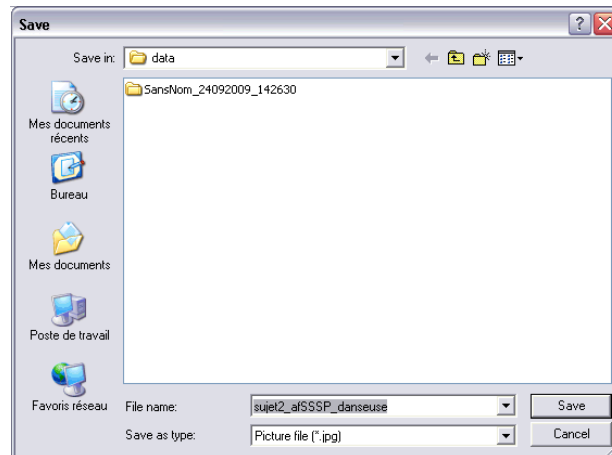
Figure 114: Taking a "snapshot" of the analysis.

A dialog box allows you to name the capture. Next, click on the "S*ave"* button to generate a JPEG picture (".jpg" extension) showing the current state of the analysis, without menus or boxes.

XIV. SAVE ANALYSIS

XIV.1.Creating a sub-analysis

(*Protocol/Sub-analysis* menu)

If only a portion of the protocol interests you, you can create a sub-analysis, in a separate file.
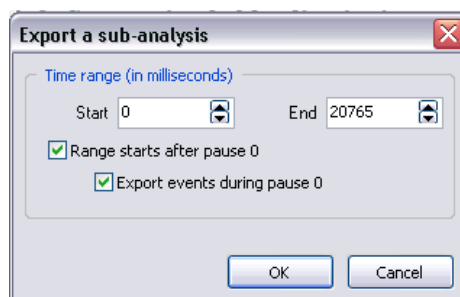
Figure 115: Defining sub-analysis time limits.

A sub-analysis is part of the overall analysis defined by a beginning and an end time (in milliseconds). Select (type in) a time for the beginning and a time for the end. If you wish your selection to start at the end of the pause "zero" (also called pre-writing pause), i.e. when writing starts, tick the checkbox "Range starts after pause 0".

Example of use: you want to export the 20 first seconds of writing, whatever the pre-writing pause duration is. Define the range from 0 to 20,000 and tick the checkbox. If you want to export the event occurring during the pre-writing pause as well, tick the second checkbox (enabled only when the range starts at 0).

Then click on the "*OK*" button. A dialog box will allow you to supply a path (directory) and a name for this new file ("*.twk*" and "*.ewk*").

If sequences exist, sequences included in the sub-analysis are also saved. If sequences "cross the border" (sequences starting before the beginning of the sub-analysis or finishing after the end of the sub-analysis), they are discarded (not saved).

XIV.2. Saving an analysis

There are two ways of saving data during an analysis:

- Click on the "*Protocol*" menu, then click on "*Save*" to save the protocol under its current name;
- Click on the "*Protocol*" menu, then click on "*Save as*". The following dialog box will then be displayed.
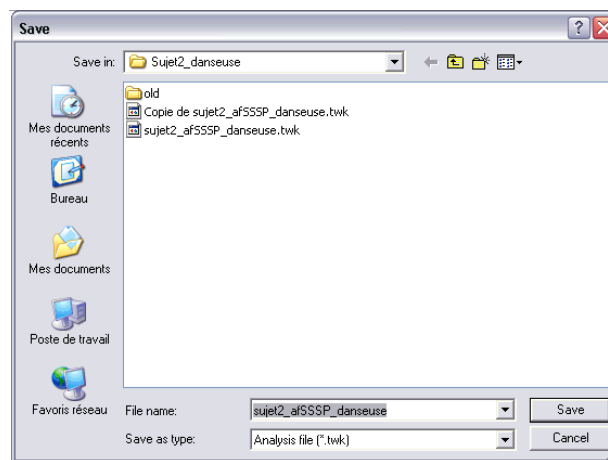
Figure 116: Saving an analysis under a new name.

Give a new name to the file (".twk") and click on the "*Save*" button. A copy of the current protocol will be created with another name.

XV. DATA EXTRACTION

XV.1. Selection of information to extract

First, you define extraction parameters: range of data, specify data details to extract and optional restrictions in the "Extraction / Preferences". You may also define parameters specific to the "Tracing dynamics" and "Visual zones" extractions.
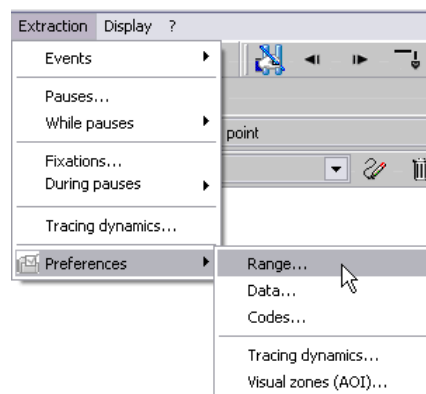
Figure 117: Extraction preference menu.

XV.1.1 Range
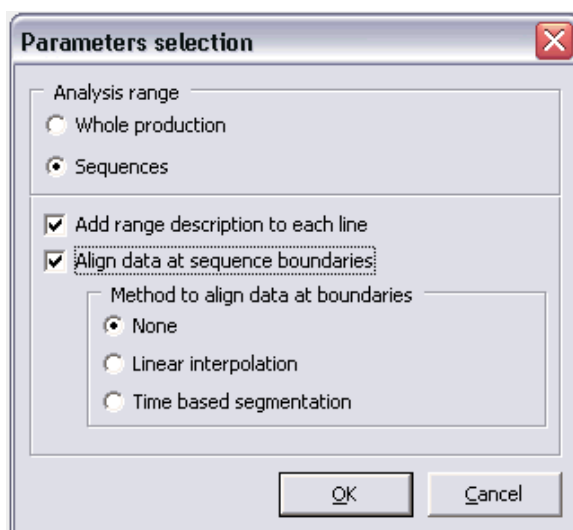
Figure 118: Select the range of data to extract.

The "***Analysis range***" frame allows you to:
- extract data from the whole protocol;
- restrict extraction to (previously defined) sequences.

Under this frame, you'll find 2 options described below:

LABEL	DESCRIPTION
Add range description to each line	Range description is composed of the sequence number (according to date of creation) the data belongs to, beginning timestamp, ending timestamp and label. If the data being extracted is not inclued in a sequence, its sequence number is "-1" and its name is "NoSeq".
Align data at sequence boundaries	Should I do anything with data beginning before and ending after a sequence? This option can only be activated if "*Sequences*" ("*Analysis range*") is selected.

Note:

If the option "Add range description" is selected and the data are extracted from the "Whole production", for each event the first matching sequence found (in the sequence list tool scrolling order, from top to bottom) is selected, even if the event belongs to more than one sequence.

If some Eye and Tab samples belongs to different sequences (e.g., a fixation begins in a sequence and continue in another sequence where a Tab event occurs), the most recent sequence will be considered.

The options of the "***Method to align data at boundaries***" frame are activated when the "*Align data at sequence boundaries*" option is ticked.

LABEL	DESCRIPTION
None	No events that are "truncated" by a sequence limit will be taken into account.

Linear interpolation	An event that is "truncated" by a sequence boundary will be included in the sequence. A linear interpolation calculation will determine the coordinates corresponding to the "cut-off-" point.
Time-based segmentation	An event that is "truncated" by a sequence limit will be partially included in the sequence. A time-based segmentation will adjust the event time to the sequence start and/or finish.

XV.1.2 Data
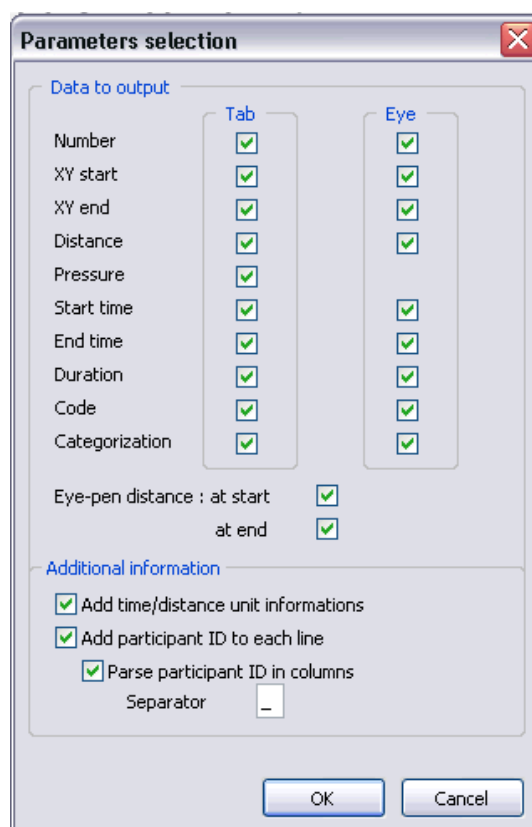
Figure 119: Select the data to output.

The "***Data to output***" frame allows you to fine-tune the details of the information you wish to extract:

LABEL	DESCRIPTION
Number	Event number in its category (Tab or Eye data) since the beginning of the protocol. The first event is numbered zero.
XY start	Coordinates when the event began.
XY end	Coordinates when the event ended.
Distance	Euclidian distance between start and finish of the event.
Pressure	Pressure of the pen on the tablet.

Start time	Time when the event began.
End time	Time when the event ended.
Duration	Event duration.
Code	Event-associated code.
Categorization	Allows you to add a categorization code to Tab and Eye data. Codes are: • *Up* (pause with pen above the tablet) or *Dn* (pause with pen pressed) or *Mov* (movement) for tablet data; • *Fix* (fixation) or *Mov* (movement) for eye tracking data.
Eye - pen distance	Euclidian distance between the position of the pen and the position of the "gaze": • when the event started • when the event ended The choice of measurement unit (tablet or metric/imperial unit) is made in the analysis configuration panel, "*Analysis*" Tab.

The "***Additional information***" frame allows you to select a label to add to each line of the extracted data:

Add time/distance unit information	Add time unit and distance unit information at the beginning of the extracted data (participant's ID is always added).
Add participant ID to each line	Add participant's name to each data line in the output file.
Parse participant ID in columns	This option can only be activated if the "*Add participant ID to each line*" has been selected (ticked). Each separator (character to be typed in the option field) found in the participant identifier name is replaced in the data extraction file with a tabulation character. The first component of the participant identifier is kept in the first column as the participant ID. The other components are placed in additional columns (labeled as *Id1*, *Id2*, etc. in the column header). <u>Example:</u> Subject 12 read "Text 2" in the "C1" condition. By naming the acquisition file "`Participant12_T2_C1`" and choosing the "`_`" ("underscore") character, the filename will be parsed into 3 columns containing "`Participant12`", "`T2`" and "`C1`". When you build data matrices (e.g. based on data extracted for several participants), it will thus be far easier to sort data based on participants, groups and experiment conditions.

XV.1.3 Codes

The dialog box allows setting codes (individual code, codes range, or codes list) to restrict extraction to data tagged with these codes.
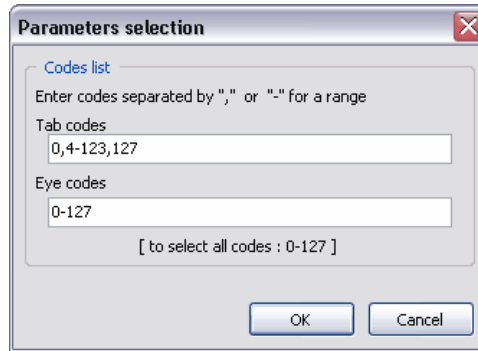
Figure 120: Define codes to set filter for extraction.

To restrict the data extraction to one or more code(s), type the code number you want in the "*Tab code*" box for the tablet data and in the *"Eye codes"* box for the eye-tracking data.

Separate isolated codes with a comma "**,**" or with an hyphen "-" if you want to select a range of codes.
Then click on the "*OK*" button to validate it.

XV.1.4 Tracing dynamics

This dialog box allows you to set the options specific to the extraction of the tracing dynamics data.
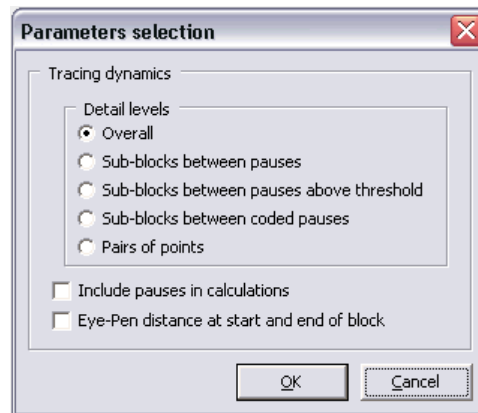
Figure 121: Tracing dynamics extraction preferences.

The parameters in the "***Detail levels***" frame are described below.

LABEL	DESCRIPTION
Overall	Gives an overall description of the pen tracing dynamics (average speed, etc.)
Sub-blocks between pauses	Pen tracing is broken up into segments, delimited by "up pauses".
Sub-blocks between pauses (above threshold)	Pen tracing is broken up into segments, delimited by "up pauses" with duration longer than the pause threshold.

Sub-blocks between coded pauses	Pen tracing is broken up into segments, delimited by "up pauses" with a significant code (>0).
Pairs of points	Calculations are made for each pair of successive points.

Below this frame (and inside the "*Tracing dynamics*" frame) you will find two options:

- "**Include pauses in calculations**". If this option is not ticked, calculations of tracing dynamics will not include pause values. They will be based on "pure" movements.

- "**Eye-pen distance at start and end of block**".

Note

For all detail levels, except "Pair of points", the pressure value extracted is a mean value, weighted with each event's duration in the considered segment, to take the possible variations of time intervals between samples into account.

XV.1.5 Visual zones (AOI)

This dialog box allows you to set the options specific to eye-tracking data extraction when in a visual zone AOI.
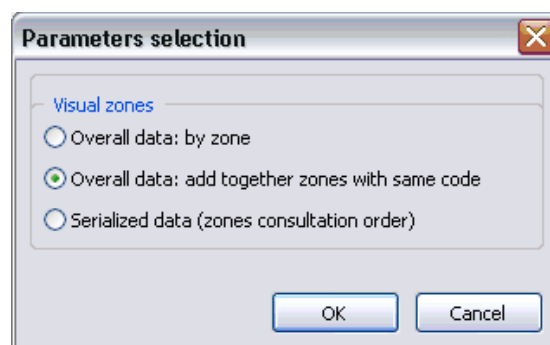
Figure 122: "Eye" data in visual zone extraction preferences.

The following table describes the parameters for the "***Visual Zones***" frame:

LABEL	DESCRIPTION
Overall data: by zone	Each line contains a summary of all the data for an AOI, in the AOI list order (by AOI number).
Overall data: add together zones with same code	Same as the above option, but if two (or more) zones have the same code for data, their data will be added together and they will be regarded as a unique zone. Each line in the file will show the summary for a particular data code belonging to one or more zones.
Serialized data (zone consultation order)	Each line of the output file shows a data summary for an AOI. The order is that of AOI consultations. Each AOI may therefore appear several times. By following the lines of the summary, you can follow the participant's "route" across the AOIs.

XV.1.6 General

This dialog box allows you to set the options that are common to all extractions.
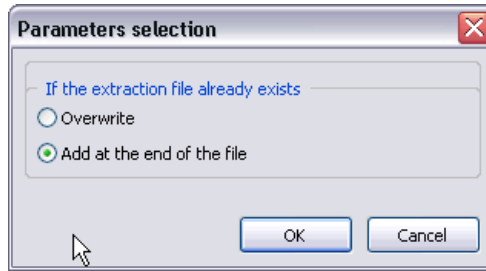
Figure 123: Overall data extraction preferences.

The radio buttons lets you set the software behavior if you selected an already existing filename for data extraction: overwrite (replace the old file with a new one) or add the new data to the existing file.

XV.2. Choose the type of events to extract

Data are "published" in tabulation-separated ASCII files (can be imported into most types of statistical software, spreadsheets and word processors).

Select the type of information you want to extract in the "Extraction" menu:

TYPE of EVENT	ACTION
All data (events)	*Extractions/Events/All events* menu.
Note: if the options "Add sequence description..." and "Whole production" are selected, it is the label of the first sequence (in the sequence tool scrolling order, from top to bottom) to which an event belongs that will be extracted, even if the event in question belongs to more than one sequence. However, if an Eye and Tab event belongs to different sequences (e.g. a fixation begins in one sequence and continues in another sequence where Tab events occur), the most recent sequence will be considered.	
Data during pauses and fixations	*Extractions/Events/During pauses & fixations...* menu.
This menu allows you to extract pauses that occur while a fixation takes place, and vice-et-versa. Remarque: There is a strong probability that some event may be noted several time in the extraction, as for example several successive fixations during a single pause. For each fixation entry, the pause values are then repeated.	
Trigger zone use	*Extractions/Events /Trigger zones (Simple)* menu
Extracts data recorded when "image" zones are used during a Simple acquisition (relies on the coordinates defined in *Configuration/Acquisition/Simple*). Extracts the Up pause before the pen is pressed in a zone, the first point in the zone (usually marking the end of the pause), the total duration of pen pressure in the zone (may sum several successive presses in the same zone), the Up pause before the pen moves out of the zone and the first press of the pen out of the zone. The start of the pause "before" and the end of the pause "after" may be used as indicators of a "round	

trip" from text to zone and thence to text. The information that is extracted depends on the tablet data extraction options (number, coordinates, timestamp, code, etc.). Note: the inclusion of pauses (an Up pause is a duration with a null pressure) in the pressure calculation will influence its result (see options for "tracing dynamics" extraction).	
Pauses	*Extractions/Pauses...* menu.
Eye (gaze) activity during pauses	*Extractions/During pauses/Eye Activity...* menu.
Fixations during pauses	*Extractions/During pauses/Fixations...* menu.
Eye activity in AOI, during pauses	*Extractions/During pauses/Eye Activity in AOI...* menu.
Fixations in AOI, during pauses	*Extractions/During pauses/Fixations in AOI...* menu.
Summary of all eye movements in AOI, during pauses	*Extractions/During pauses/Eye Activity in AOI summary...* menu.
Summary of all fixations in AOI during pauses	*Extractions/During pauses/Fixations in AOI summary...* menu.
Fixations	*Extractions/Fixations...* menu.
Eye activity in AOI	*Extractions/Visual zones (AOI)/Eye activity...* menu.
Fixations in AOI	*Extractions / Visual zones (AOI) / Fixations...* menu.
Eye activity in AOI summary	*Extractions/Visual zones (AOI)/ Eye activity summary...* menu.
Fixations in AOI summary	*Extractions/Visual zones (AOI)/fixation summary...*menu.
AOI scanpath summarized as a string of letters	*Extractions / Visual zones (AOI) / Route as string* menu
Extracts the route of fixations through the AOI as a string of letters, each letter corresponding to a zone: A=zone 1, Z=zone 26. Only the first 26 AOIs are taken into account (beyond the "alphabet scope", AOIs are labeled with "?"). You may get something like "`GAHGADCDEGFGABGAGFGAGAGEEGACHGH`"... This sort of data can be used to compute the likelihood of different "routes" (e.g. with the Levenshtein distance) or to perform pattern matching analyses.	
Tracing dynamics (pen movement data)	*Extractions/Tracing dynamics...* menu.

Lastly, a dialog box allows you to give a name (and optionally a new path) to the text file that is to contain the data.
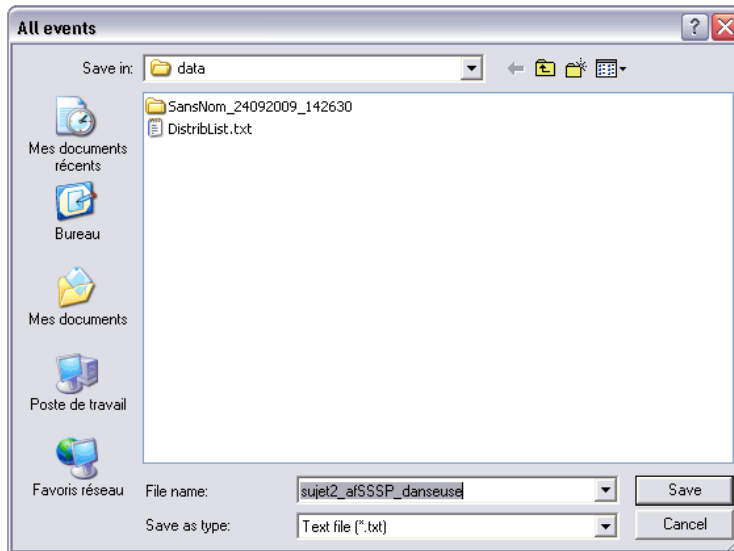
Figure 124: Naming the output file.

Click on the "*Save*" button to complete the process.

XVI. DATA EXPORTS

XVI.1. Exporting data as a text file

(*File/Exports/Text file...*menu)
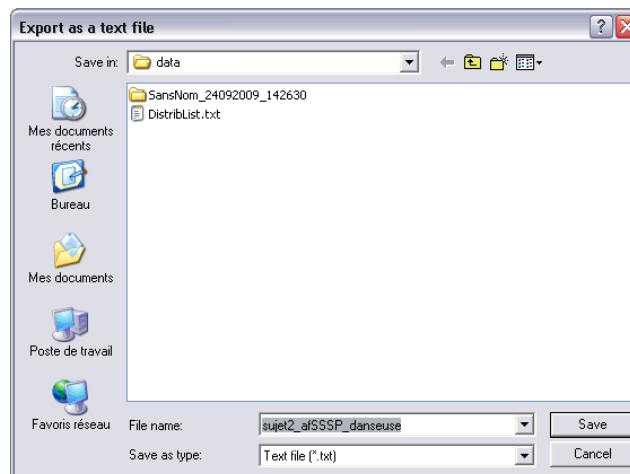
Figure 125: Naming the text file.

Select a directory, choose a name for the file, and click on the "*Save*" button.

Let us suppose that you gave *MyFile* as the filename. In this case, tablet data will be exported to *MyFile.tab.txt* and eye-tracking data to *MyFile.eye.txt.*

Tablet data include time (in milliseconds), X and Y coordinates, and pen pressure.

Eye data include time (in milliseconds) and X and Y coordinates.

XVI.2. Exporting data as a G-Studio file

(*File/Exports/G-Studio...* menu)

Select a directory, choose a name for the file, then click on the "*Save*" button.
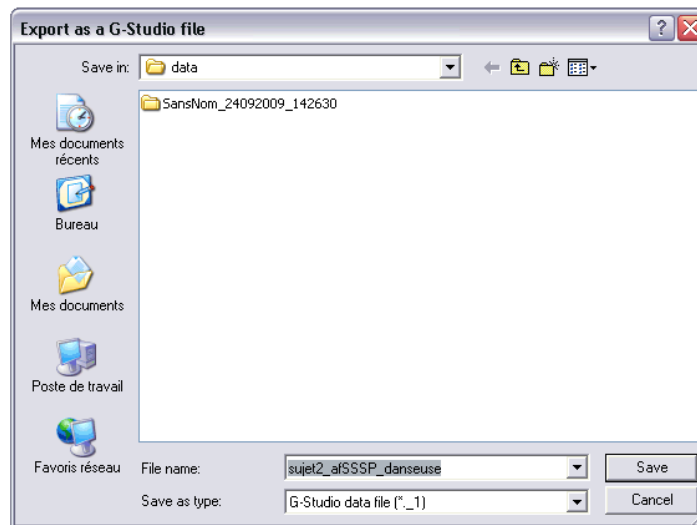
Figure 126: Naming the G-Studio file.

Let us suppose you gave *MyFile* as the filename. Tablet data will therefore be exported to the file named *MyFile._1*.

CHAPTER 4: other functions

I. KEYBOARD: SHORTCUTS, RECORDER AND AUTOMATION

I.1. Menu accelerators and keyboard shortcuts

In Eye and Pen, each menu item has an accelerator, i.e. a letter is underlined in the menu item name that shows which key of the keyboard should be pressed to activate this item. Theses accelerators are visible since the menu has been activated with the help of the keyboard ALT key.

The Eye and Pen menu also handles keyboard shortcuts, i.e; keys combinations that allows you to directly activate a menu item, without any need to open the menu first or scroll down any sub-menu.

Default shortcuts are listed below:

KEY(S)	DESCRIPTION
CTRL + N	New analysis
CTRL + O	Open an existing analysis
CTRL + S	Save the current analysis
CTRL + F4	Close the analysis
ALT + F4	Quit Eye and Pen
SHIFT + E	Re-center the analysis "window" to make the eye cursor visible
SHIFT + T	Re-center the analysis "window" to make the pen cursor visible
HOME	Jump to start of protocol
END	Jump to end of protocol
Up arrow	Previous pause
Down arrow	Next pause
Left arrow	Previous data (event)
Right arrow	Next data (event)

Eye and Pen's menu keyboard shortcuts are customizable.
Select the "*Display / Menu shortcuts*" menu.
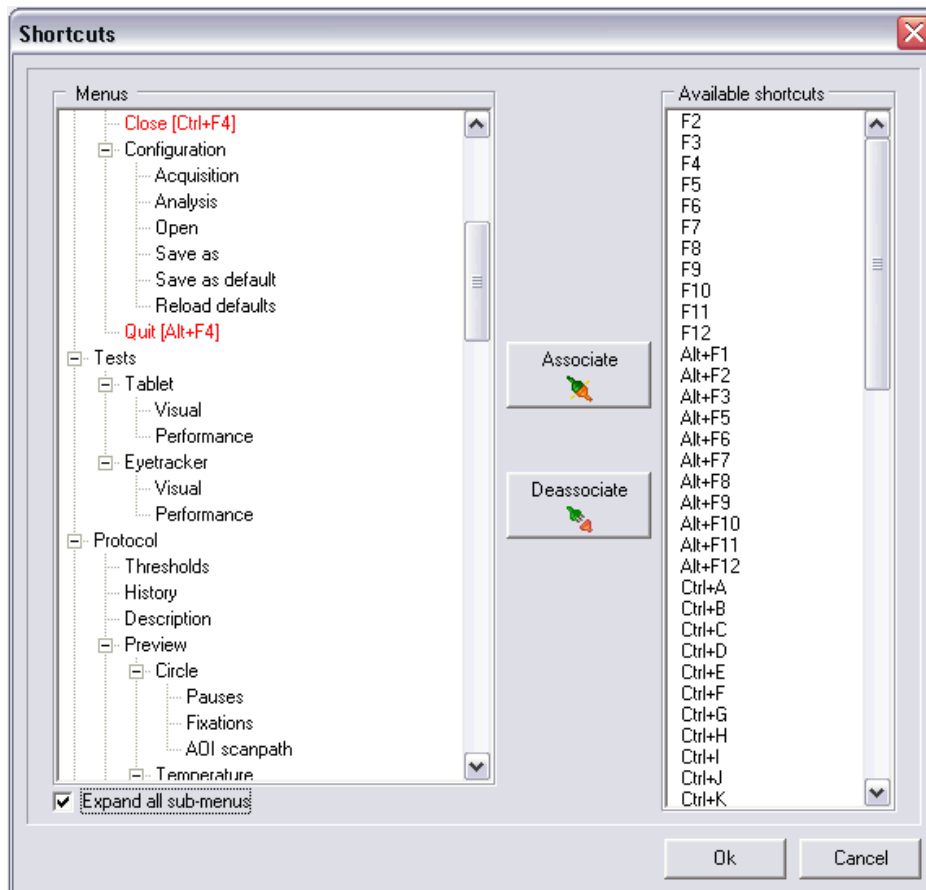
Figure 127: Setting/unsetting shortcuts for menus.

The left-hand column shows Eye and Pen menus and submenus, and the right-hand column shows available shortcuts.

To associate a shortcut with a menu item, select the menu item in the left-hand column, select the shortcut you want in the right-hand column and click on the Associate button. The shortcut will disappear from the right-hand column and the menu item will be updated in the left-hand column.

To dissociate a shortcut, select the relevant menu in the left-hand column, and click on the Dissociate button. The shortcut will once more be displayed in the right-hand column.

To reset shortcuts to their default values, close Eye and Pen and delete the file named "shortcuts.MSL" in the Eye and Pen directory. Start Eye and Pen.

I.2. Keyboard recorder

(*Display / Keyboard recorder* menu to display/hide the tool)

The keyboard recorder is intended to record keyboard keys and/or keypress combinations when performing an analysis in Eye and Pen. This recording can later be reused by the EPKeys utility (p. 174) to replay keypress sequences at will. This makes it possible to automate some of the protocol processing or data extractions and even to repeat the processing of a whole set of protocols.

Figure 128: Keyboard keypress recorder.

To activate or deactivate the recorder, click on the button in the "*Keyboard recorder*" toolbar.

Figure 129: The recorder is switched on

When you stop the recorder, a dialog box will appear and offer to save the list of keypresses in a text file.
Select the destination folder, give the file a name and click on the "*Save*" button.
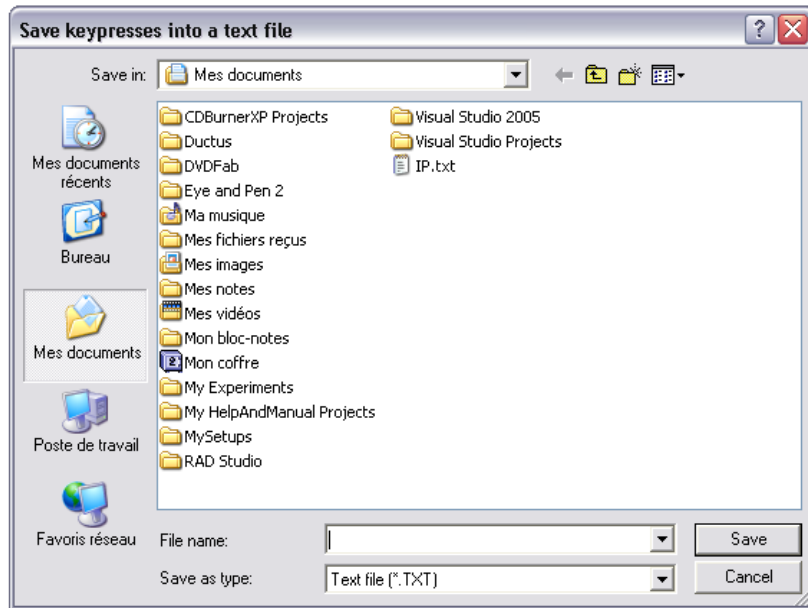
Figure 130: Saving keypresses in a text file.

The text file will contain all the keys or keypress combinations while in the Eye and Pen analysis mode, with one key (or combination of keys) per line.
Key combinations and "special" keys are written between braces (e.g. ALT+f is written as {ALT+f}, and the escape key is represented by {ESC}).
The file can then be used within the EPKeys utility.

<u>Note</u>: shortcuts and accelerators (letters underlined in a menu item name indicating which key to press to activate the item) are not systematically the same in the French and English versions of Eye and Pen.

I.3. EPkeys

EPKeys is an external and independent utility (EPkeys.exe) which can drive Eye and Pen.
It can simulate keyboard keypresses and send them to Eye and Pen, as if you were actually pressing on keys in Eye and Pen.

This allows you to "script" tedious and repetitive tasks, and let the computer do all the work for you, instead of spending hours repeating the same sequences over and over again.

The primary function of EPKeys is to use a file of keypresses recorded in Eye and Pen to reproduce processing sequences or data extraction in one or several protocols.

Since you can choose the software targeted by EPKeys, you can use it to drive other software. But please remember that it was neither built nor tested for this purpose, so you will be doing it at your own risk.
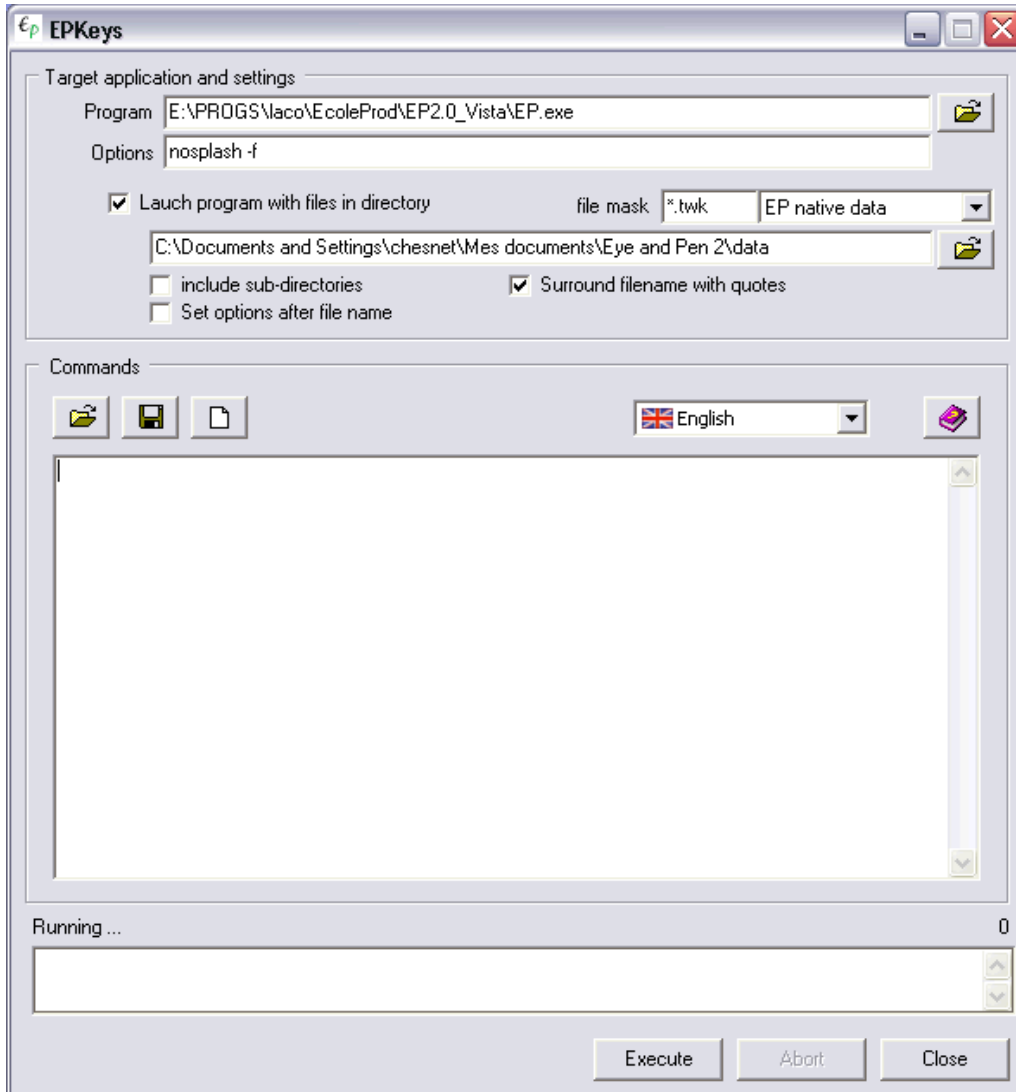
Figure 131: EPKeys.

The fields of the "*Target application and settings*" frame allow you to define the software to drive and how this should be done:

LABEL	DESCRIPTION
Program	Shows the path to the target program (*.exe). If needed, click on the ![folder] button to select a new one.
Options	Type in the options that will be passed via the command line to the target program (see p. 19 for the Eye and Pen options). "`nosplash -f`" is usually a good option for Eye and Pen.
Launch program with files in directory	If this checkbox is ticked, the software will be launched for each file in the selected directory. You do not need to open each data file, as this will be done automatically.

File mask	Defines the filename extension to look for. For Eye and Pen, this may be "*.twk", the analysis filename extension, or "*.tab" the acquisition data filename extension.
Include sub-directories	If this option is selected, the search for files will be recursive, in that it will also be performed across all the subdirectories of the selected directory.
Set options after filename	This option defines whether the "sentence" option should be placed before or after the filename in the command line. If this option is activated, the options will be written after the filename. For Eye and Pen, this option should not be ticked.
Surround filename with quotes	If this checkbox is ticked, the filename added to the command line will be "framed" with quotation marks. This can avoid problems with filenames that include spaces, for example.

The "*Commands*" frame allows you to define the keyboard keypresses to simulate, save them in a file, load them from a file, etc.:

LABEL	DESCRIPTION
	Click on this button to open and load a text file containing commands (keyboard keypresses).
	Clicking on this button allows you to save the content of the "Commands" memo in a text-only format file.
	This button allows you to clear the "Commands" memo of its content. If there is any content, EPKeys will offer to save it in a text file.

Writing commands

Simply write down the alphanumeric character corresponding to the key you wish to activate, one key (or combination) per line. For example, to simulate a press on the "e" key, just write, well... "e"!
EPKeys is case-sensitive: it differentiates between uppercase (capital) and lowercase letters. This means that a letter such as "E" will be interpreted as the combination of the "Shift" and "e" keys. So be careful about menu shortcuts: when in doubt, use lowercase letters. For example, the combination of the "Ctrl" and "n" keys (usual shortcut for the "*File / New*" menu) will not work if you write "Ctrl+N", because the combination will be interpreted as "Ctrl+Shift+n".

Comments

You can add comments if you begin the line with ";" or "//".
For example:
; is a comments line
// this is another comments line, choose your style ;-)

Special keys

Special keys are those that do not represent an alphanumeric character. They should be written between braces. EPKeys recognizes the following special keys:

{ALT}	Alt key
{CTRL}	Ctrl key
{SHIFT}	shift (capital) key
{TAB}	tabulation key
{BAK}	backspace key
{ESC}	escape key
{RET}	return key (enter)
{SPC}	spacebar
{END}	end key
{HOME}	home key
{LEFT}	left arrow key
{RIGHT}	right arrow key
{F1}	F1 key
{F2}	F2 key
{F3}	F3 key
{F4}	F4 key
{F5}	F5 key
{F6}	F6 key
{F7}	F7 key
{F8}	F8 key
{F9}	F9 key
{F10}	F10 key
{F11}	F11 key
{F12}	F12 key

Key combinations (mixed keys)

The CTRL, ALT and SHIFT keys can be combined with common keys to write shortcuts such as {ALT+a}, {CTRL+n} (usually used for the "*File / New*" menu) and {CTRL+SHIFT+e}.

Menu shortcuts

You can find the shortcuts associated with menu or submenu items in front of their name in the application that you wish to "drive".

As a rule of thumb, you can activate (open, scroll down), the main menu by pressing the "Alt" key.

Menus and submenus that do not have shortcuts sometimes have menu accelerators: a letter is underlined in the menu name to show which key of the keyboard you need to press to activate this menu or submenu.

Close the program

If you launch your commands recursively, for example with all the files contained in a directory, do not forget to end your command list when you close the program, otherwise you may get into trouble. In most of the applications, the {ALT+F4} combination is the shortcut to "Quit".

Configuration file

The EPKeys configuration and parameters are saved in a text file named EPKeys.ini.
The parameters in the [User configuration] section are modified via the graphic interface of EPKeys (directory names, options, etc.)

A special section named [Timeouts] contains some "hazardous" parameters that can be edited "by hand".

PARAMETER	DESCRIPTION
IdleTimeOut	Represents the number of milliseconds of CPU inactivity beyond which the driven application is deemed to be idle. Increasing this value will slow down the pace of sending keypresses, but can help to avoid problems related to function execution timings. As a safety precaution, EPKeys only accepts values within the 20–5,000 range (default value: 200).
UpTimeOut	Represents the number of milliseconds EPKeys should have to wait for the "driven" program to start and be ready to receive keypresses. As a safety precaution, EPKeys only accepts values within the 500–10,000 range (default value: 1,000).
CloseTimeOut	Represents the number of milliseconds EPKeys should have to wait after the last keypress of the list has been sent. This parameter is very important when one wants the program to "loop", for example to repeat a series of keypresses for all the files in a directory. If EPKeys does not allow enough time for the "driven" program to close, it will try to launch the program again before it can close, which can lead to problems. Eye and Pen is closed with the usual {ALT+F4} key combination ("*File / Quit*" menu). As a safety precaution, EPKeys only accepts values within the 500–10,000 range (default value: 1,000).

Before using EPKeys: a few recommendations

1. Avoid complex scripts. If you launch the software to process a set of files, bear in mind that EPKeys is a blind robot: if something unforeseen occurs (e.g. a confirmation panel to check), EPKeys will continue to send keypresses without taking into account the fact that its keypress list no longer fits (it has no way of knowing). It is therefore a good idea to perform some manual testing (doing what EPKeys is supposed to do) before launching EPKeys. Try to have only one simple task executed at a time.

2. Test your script on a single file to make sure that everything goes according to plan before launching it for dozens of files.

3. The *File | Configuration | Analysis* menu, *Application* tab allows you to avoid dialog boxes such as "Warning: this file already exists…".

4. Remember that Eye and Pen will be using its own default configuration. For example, the default output directory for extractions will be the last directory used for an extraction in Eye and Pen. So, if you want data to be extracted from a new (or dedicated) directory, create this directory (e.g. with Windows Explorer) and perform the first extraction "by hand" in Eye and Pen, in order to select this new directory as the output directory.

5. If you are launching commands on for a number of files, do not forget to close the application launched by EPKeys at the end of your list of commands ({ALT+F4} generally does this). This is necessary because some applications (including Eye and Pen) will not start twice in parallel. If a software is already running, it will be re-activated and EPKeys will not be able to communicate with it because EPKeys is supposed to communicate with a newly created software instance, not the old one.

II. CUSTOMIZING YOUR WORKSPACE

When performing an analysis in Eye and Pen, the environment in which you work can be modified to suit your needs or to be more comfortable.
This environment is saved and reloaded with Eye and Pen.

II. 1. Move toolbars.

Click (mouse left button) on the handle (sort of vertical line) to the left of the chosen toolbar and drag it to where you want it to be. You can even drag it under the last toolbar (at the bottom), as shown below. The toolbar "docking station" can enlarge itself (add a line at the bottom).

Figure 132: Dragging a toolbar.

Release the mouse left button when the desired location has been reached.
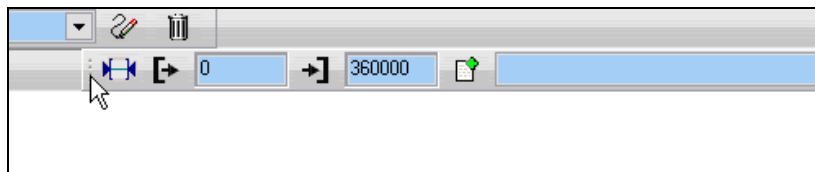
Figure 133: Toolbar moved.

II. 2. Make toolbars "float".

Click (mouse left button) on the handle to the left of the chosen toolbar and drag it toward the bottom of the screen. Do not release the button until the toolbar has been "extracted" from the toolbar dock.
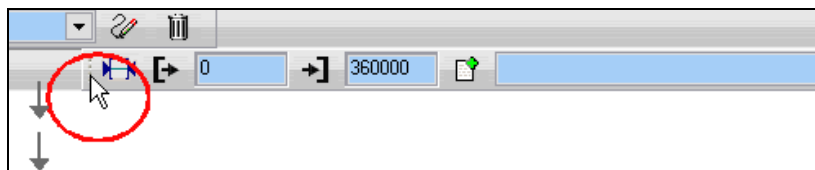
Figure 134: Undocking a toolbar.

The toolbar will now be inserted into a new window.
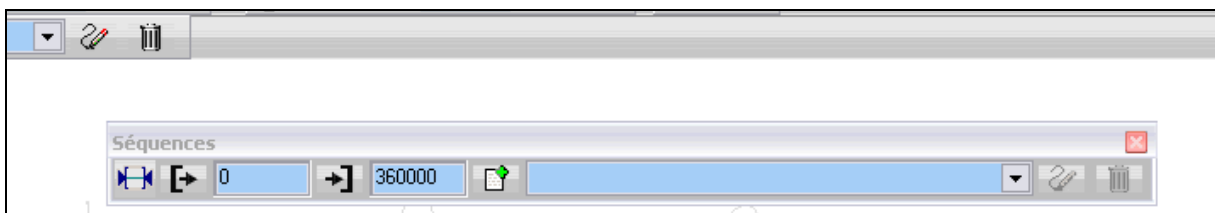
Figure 135: A floating toolbar.

You can move this window wherever you like in the screen.

Then again, if you want to "dock" the toolbar, just drag it back to the toolbar "docking station".

Figure 136: Docking a toolbar.

II. 3.   Show / hide toolbars.

You can show or hide toolbars using the *Display | Toolbars* menu.
Click on a toolbar name to show it if it is hidden, or to hide it if it is visible.
A visible toolbar has a "Check" mark in front of its name in the menu.

II. 4.   Customize toolbars.

You can customize toolbars using the *Display | Toolbars | Customize* menu.
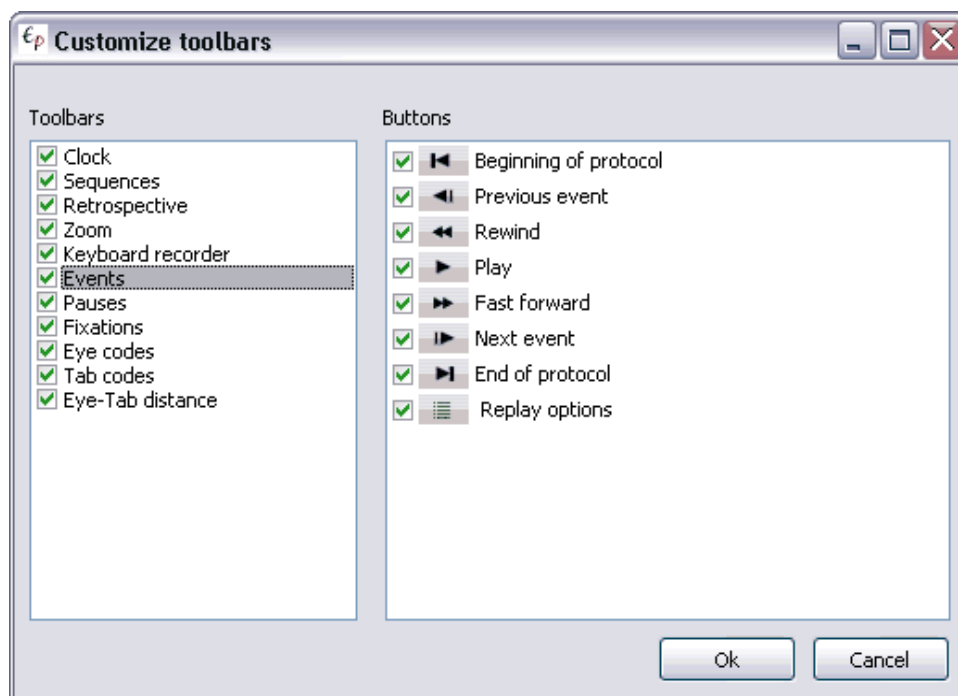
Figure 137: Toolbar customization panel.

The list to the left shows the available toolbars. Check or uncheck the checkbox in front to show or hide the toolbar.

To select a toolbar without modifying its visibility, click on its name. Then, the right-sided list will be updated to show the buttons and tools located on this toolbar. Like for the left-side

list, the checkbox reflects visibility of the buttons/tools. Checked (ticked) means the button is visible. To modify this, check or uncheck the checkbox.

Click on the "*OK*" button to apply all changes or on the "*Cancel*" button to discard changes.

Hint: when playing a lot with customization, some toolbar resizing trouble may happen (toolbar looks weird). Undocking and docking again the toolbar refreshes its look.

II. 5. Resize the "palette".

First of all, the palette can be virtually hidden if you double-click on its title bar.
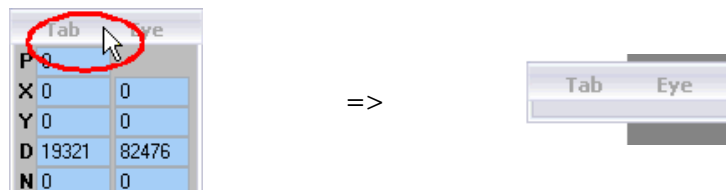
Figure 138: Shrinking the palette.

Double-clicking again reverses the process.

Next, you can resize the list of codes.
Just click and drag the palette's lower edge (border), either up to shrink the list, or down to enlarge it.
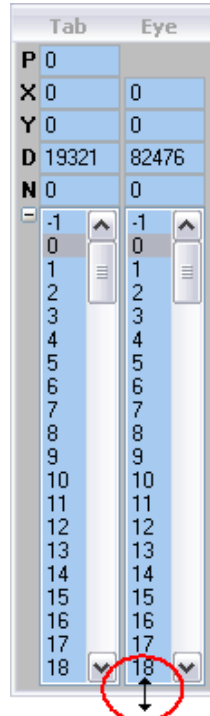
Figure 139: Resizing the code list.

You may also prefer to hide the whole list. This is easy: just click on the "minus" button. The button will then become a "plus". Simply click on it again to show the list.
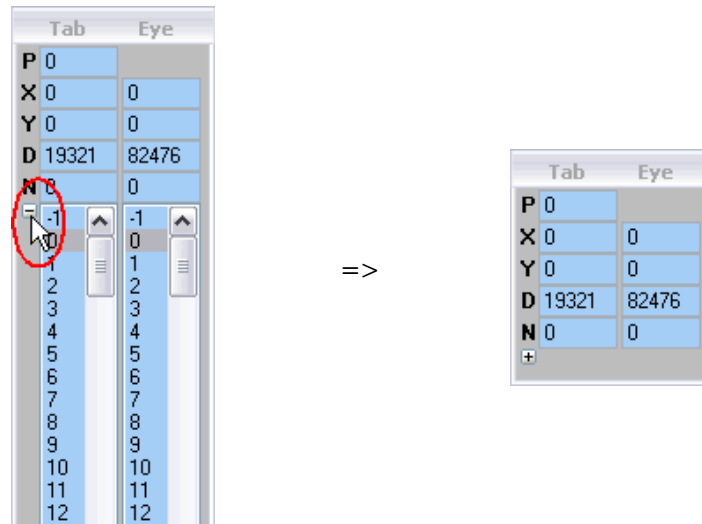
Figure 134: Hiding the code list.

III. MODIFY DISPLAY RATIO

Using the *Display* menu you can either enlarge or shrink the protocol display size.
Clicking on the *Display | Enlarge* menu doubles the size of the protocol on the screen.
The *Shrink* submenu does the opposite.

Remember that enlarging the protocol size requires additional video memory, so trying to enlarge too much may cause display problems (restarting Eye and Pen solves the problem).

Also remember that some of the Preview functions use pixel sizes as a reference (e.g. for fovea size projection on display calculation). You may therefore need to recalculate fovea size to adapt the preview to this new display ratio (*Configuration | Preview* menu).

IV.   UNIT CONVERSIONS

IV. 1. Angle - pixels

(*? / Conversions/ Angle -> Pixels* menu)

This dialog box helps to convert, for example, visual angles into screen pixels.
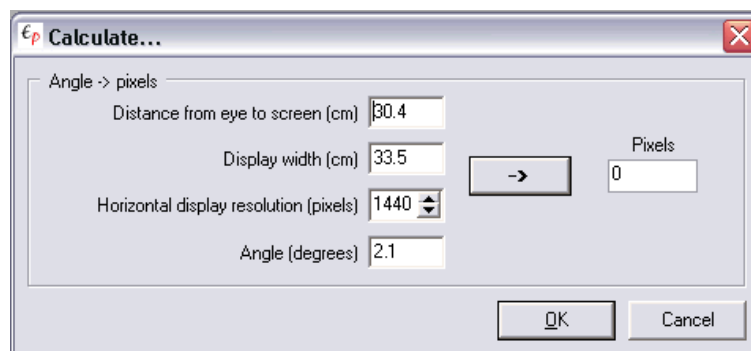
Figure 141: Angle (degrees) to pixels conversion.

Fill the boxes with the requested values and click on the "[->]" button.

Value in pixels is displayed in the right side "Pixels" field of the dialog box.

IV.2. Tablet units - distance

(*? / Conversions/ Tablet units <->distance* menu)

The "Eye and Pen" software has a tool to help you c*onvert* tablet units (lines) into distance measurement units (centimeters or inches, according to the tablet version).
The following dialog box is displayed.
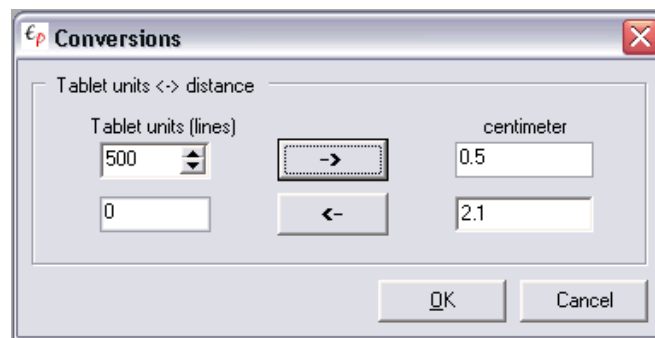
Figure 142: Tool to convert Tab units to distance units.

To convert to distance units (e.g. into millimeters), type your value in the "*Tablet units*" box, then click on the right arrow "[->]" button.

The value in measurement units is updated in the box to the right (in this case, the "*centimeters*" box).

To convert to lines, type the value in the box on the right of the dialog box (bottom line), then click on the left arrow "[<-]" button.

The value will be updated in the box on the left.

IV. 3. Tablet units – pixels

(*? / Conversions/ Tablet units <->pixels* menu)
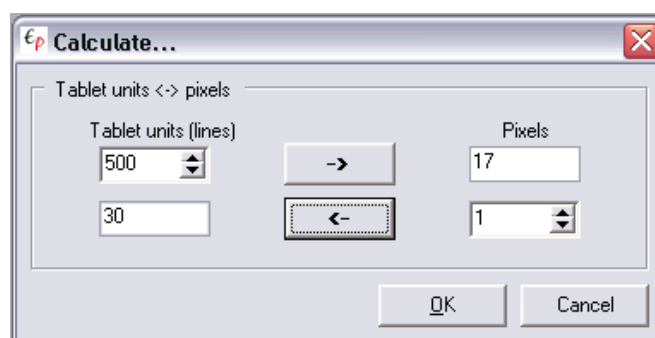
Figure 143: Tablet units to pixels conversion.

To convert to pixels, type your value in the "*Tablet units*" box, then click on the right

arrow "→" button. The value in measurement units is updated in the box to the right (in this case, the "*centimeters*" box).

To convert to lines, type the value in the box on the right of the dialog box (bottom line), then click on the left arrow "←" button. The value will be updated in the box on the left.

V.   CALIBRATION TEST FOR EYEPUTER

As Eyeputer does not have an interface for a calibration or drift test procedure, a specific calibration test panel has been developed.
To launch this panel, use the "TestCalibration" or "TestDrift" command in a script (for information about using commands, see Part III. Script-based acquisition).
Once a script-based acquisition has been launched and one of these two commands sent, you will see the following panel:
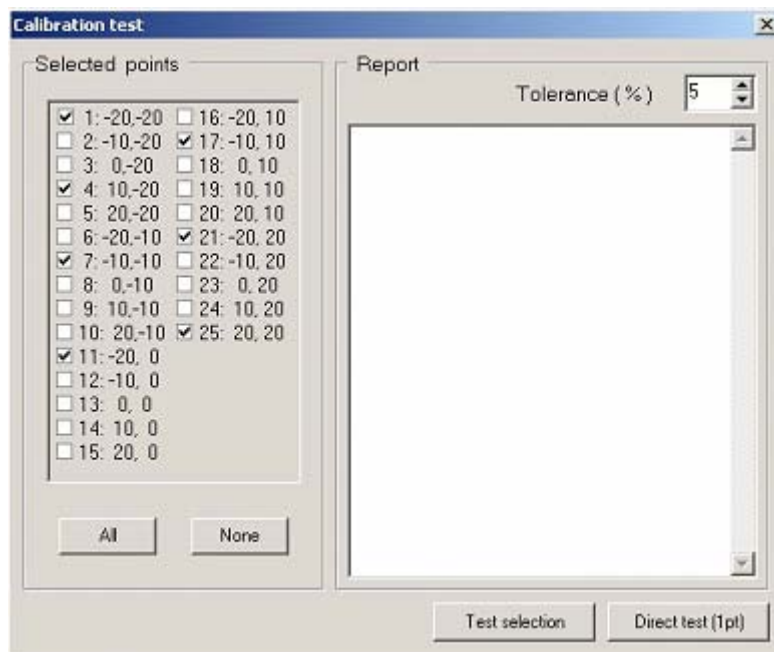
Figure 144: Calibration test panel.

The calibration test panel for Eyeputer shows two frames, labeled:
- "Selected points": select the points that will be used in the test (tick the boxes);
- "Report": shows the coordinates returned by EyePuter, indicating the deviation relative to the theoretical coordinates, and an indication if the deviation exceeds the tolerance limit.

To fine-tune this threshold (percentage), change the value of the parameter located to the right of the "*Tolerance (%)*" label.
The reference for maximum deviation is the distance between the first point of the first line and the second point of the second line of the calibration grid (no.1 and no.7).

The "*Test selection*" button located at the bottom of this frame allows you to test the whole set of selected points.

The "*Direct test*" button allows you to continuously test the first selected point (coordinates returned by the EyePuter are displayed one after the other until you cancel the test).

Press on the keyboard space bar to validate the moment where the participant is looking at the point.
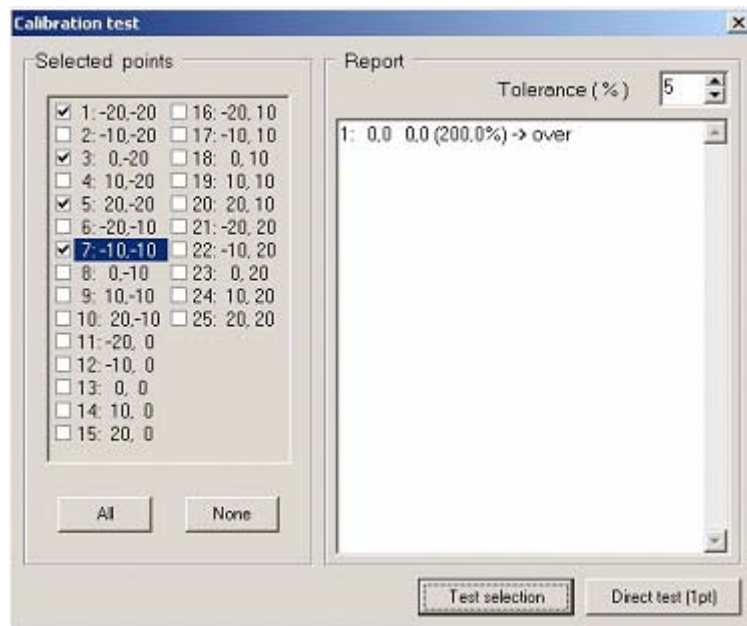
Figure 145: Calibration test report.

In the above example, the points 1, 4, 7 and 11 have been tested and, each time, the returned coordinates were 0 on X and 0 on Y (eye tracker is stopped). Eye & Pen indicates (between brackets) the percentage of deviation between the "theoretical" coordinates and those that were actually received. If this distance is above the tolerance threshold, "-> over" is added.

VI.   SYSTEM INFORMATION

(*?/System information* menu)
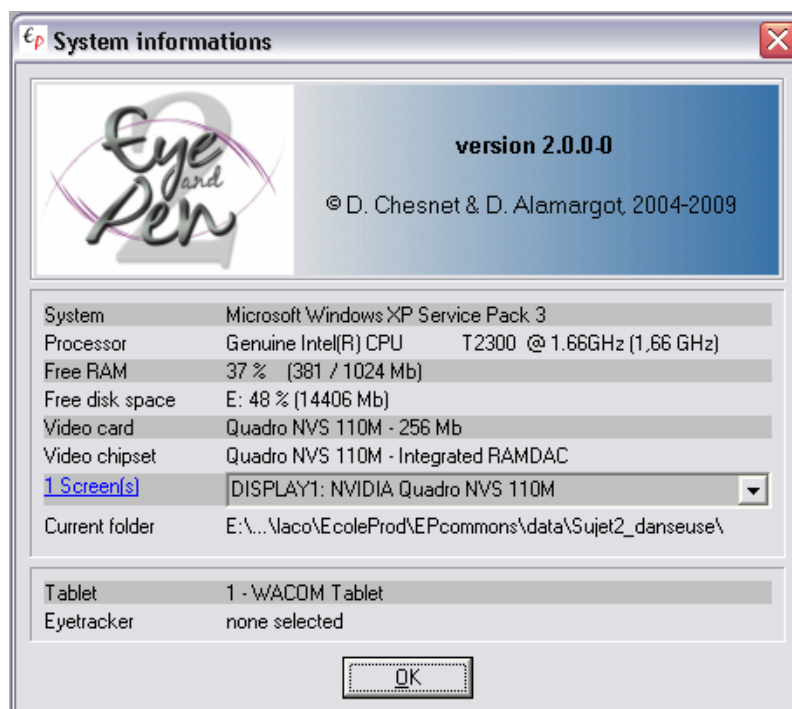
Figure 146: System information dialog box.

A click on the "Screen" link opens the Display feature of the Windows control panel.

VII. LEGAL INFORMATION AND ACKNOWLEDGEMENTS

(*?/About* menu)
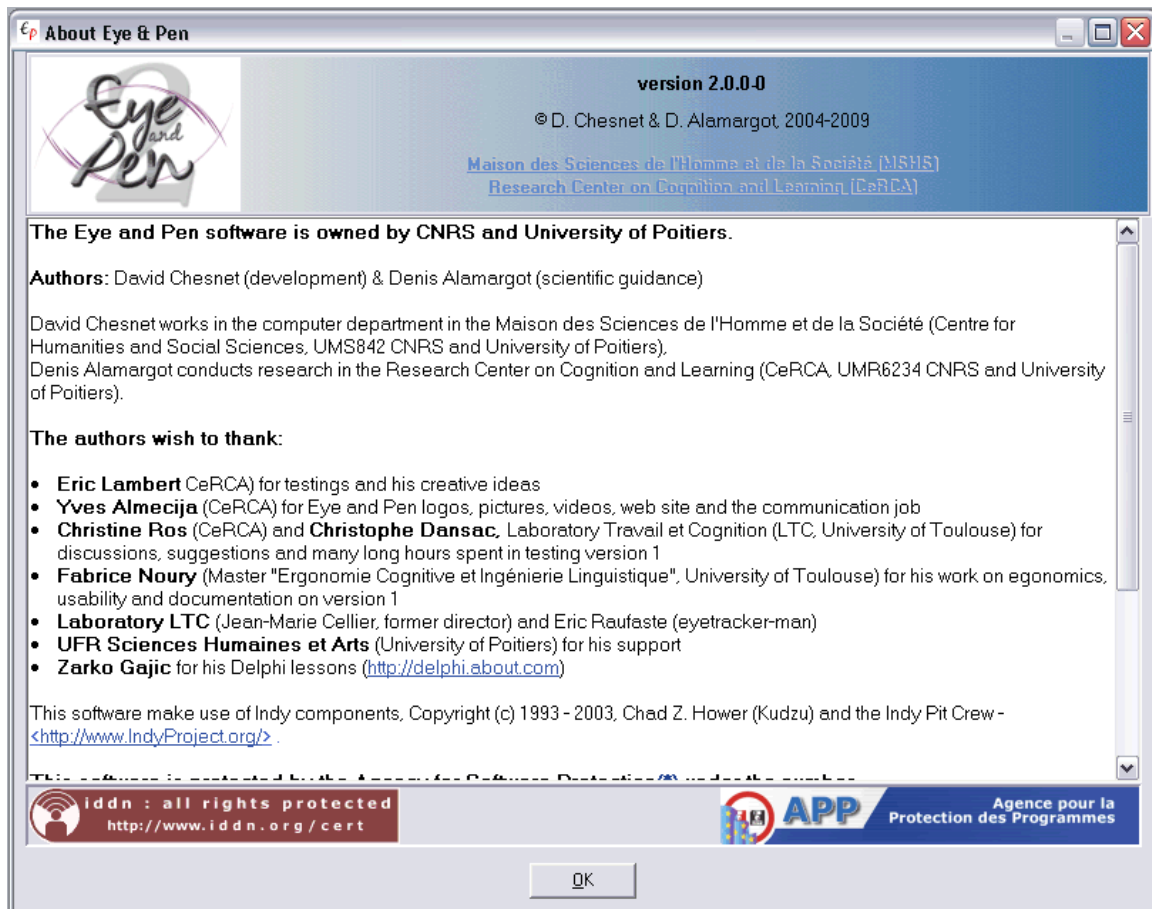
Figure 147: Information and acknowledgements dialog box.

The "http" links in the text can be clicked. They open the Internet site mentioned (as far as you have a working connection).

If you click on the "Eye and Pen" picture, you'll get to the http://www.eyeandpen.net web site.

CHAPTER 5: appendices

I.   NETSYNC WITHIN A WI-FI NETWORK

Here we present a configuration we used for a word copy task, managed in a group.
NetSync was used to perform the data collection for the experiment with a group of 5
participants, with a single Eye and Pen license.
Only general information is given below, and the specific implementation may vary according
to the systems and devices that are used (refer to their own documentation).

The system comprised:

- 5 laptop computers (clients), each with a tablet, an internal Wi-Fi adapter and an Eye
  and Pen installed, without dongle;

- 1 laptop computer equipped with an internal Wi-Fi adapter, an external hard disk (to
  receive the clients' data) and Eye and Pen software with dongle;

- 1 Wi-Fi 802.11n router (300 Mb/s).

The main advantage of this configuration lies in his "mobility": it can be operated almost
anywhere, including schools, because it is easy to set up and dismantle.

Figure 148: NetSync within a Wi-Fi network.

I.1   Wi-Fi router configuration

In order to block undesirable connections, the Wi-Fi router is configured as a semi-static DHCP: the IP address list is entered "by hand" in the router Web interface. Each of these addresses is subsequently allocated to a particular host (clients or master). The MAC address of the host's Wi-Fi adapter is added to each IP address.

The filtering is then activated on the router: only the hosts whose MAC address has been entered in the DHCP can connect to the network.
Thus, no hosts other than client and master hosts can join our "private" network.

I.2   Host configuration (clients and master)

The Wi-Fi alone is activated and all other communication devices should be deactivated (Ethernet, Bluetooth, etc.).

The host's IP address is set to a dynamic address (DHCP). Its machine name is the same as the one that has been typed in the Wi-Fi router DHCP configuration.

Hosts are configured to automatically connect to our network and no other (this can be stipulated in the Wi-Fi adapter configuration, see Network connections in Windows).

I.3   Script

Only small changes need to be made to scripts.
The following sequence (added at the beginning of the script) makes the participant wait for the session to start:

```
DisplayMsg(Wait for the session to start...,-1,-1,-1,FALSE)
WaitForNetSync
HideMessage
```

Some other specific commands (see Netsync commands, p. 81) can either be inserted into strategic locations in a script, such as pause between blocks of items, or wait until all the participants have finished their trial to start a new one.

II.   CONFIGURING THE TABLET WITH A DOUBLE SCREEN

A double screen configuration is a display configuration (defined in Windows) where you work with a computer monitor <u>and</u> an LCD tablet.

With this kind of configuration, the Windows operating system will consider that the tablet screen and the computer screen are merged into single desktop.

To avoid having the tablet coordinates mapped to both screens, check in your tablet's configuration panel that the tablet is mapped to the correct monitor (not the whole desktop).

III   SCRIPT EXAMPLES

This section of the manual is intended to help you learn/practice script writing based on practical cases.

III.1. Have a task re-done

This script allows the participant to start task 1 a maximum of four times and task 2 a maximum of two times.
Each task is triggered by a zone on the tablet (we will call the zones' coordinates on the tablet X1, Y1, X2 and Y2, but in a "true" script you must replace these symbols with values expressed in tablet units).
We want to record the participant's activity on each occasion, i.e. in a different acquisition file.
When the maximum is reached for a given task, if the participant presses the relevant zone again, nothing will happen. When he or she presses the pen in the zone associated with "Stop", the script will jump to the label named "Next".

```
;       we define three zones on the tablet, each being associated with a label
DefineTabZone(x1,y1,x2,y2,Task1)
DefineTabZone(x1,y1,x2,y2,Task2)
DefineTabZone(x1,y1,x2,y2,Stop)
JumpTo(Go)

:Task1
  JumpToIfNumberIs(Go,5,FALSE)
  OpenRec(_task1_%I%)
 ;       do something
  JumpTo(Go,TRUE)

:Task2
  JumpToIfNumberIs(Go,3,FALSE)
  OpenRec(_task2_%I%)
 ;       do something
JumpTo(Go,TRUE)

:Stop
  JumpTo(Next)

:Go
  WaitForTabZones(TRUE,TRUE,TRUE)

:Next
  CloseRec
;  Script continues
```

Note: if the acquisition file is already closed, a call to CloseRec will have no "negative" effect.

III.2. Mask production (Simple)

This script allows the participant to write a page of text, but the display on the screen will be masked by a plain color (green). The participant will only be able to see what he or she writes in a round zone around the pen tip location (this zone keeps track with the pen tip's position).

Figure 149: Masking production.

; select the unmasking file, the one that will allow you to see "through" the mask covering
; the display on screen
```
SetUnmaskFile(..\cursors\Mask_circle120.bmp)
```

; masking mode no.1: the display is filled with a plain color
```
SetMaskingMode(1)
```

; filling color: green
```
SetMaskingFillColor(#00FF00)
```

; activate display masking
```
ActivateFeedbackMasking(TRUE)
```

; start recording
```
RecStandard
```

The unmasking file is contained in the *Cursors* folder, not the *Stimuli* folder, which explains why a relative file path is used here: "`..\cursors\Mask_circle120.bmp`" (for further explanations, see p. 15).

III.3. Mask production with unmasking locations

The purpose of this script is to have a participant write 5 words in "slots", whose display on screen will be masked. The participant will only be able to see the word he or she is writing, together with the previous word.

;   unmasking file selection
```
SetUnmaskFile(masque_5_mots.jpg)
```

;   masking mode n°2: a picture file is displayed on screen and covers the production.
```
SetMaskingMode(2)
```

;   masking file selection
```
SetMaskingFillPic(fond_blanc_5_mots.jpg)
```

;   define unmasking zones (they match with the "slots" of the masking file
;   from left to right.
```
DefineUnmaskZone(35,350)
DefineUnmaskZone(200,350)
DefineUnmaskZone(365,350)
DefineUnmaskZone(530,350)
DefineUnmaskZone(695,350)
```

;   rule: the participant will be able to see the content of the zone he's in and the left side one
```
SetUnmaskZonesRange(-1,0)
```

;   going back is forrbidden: the participant will not be able to unmask a zone
;   previously "visited"
```
SetUnmaskZonesBackTracking(FALSE)
ActivateFeedbackMasking(TRUE)
```

;-----**Trial n°1**
```
WaitForKeyPressPic(croix.bmp,-1,-1)
```

;   Hint: we use the same picture as a background and as a mask
```
DisplayPic(fond_blanc_5_mots.jpg,-1,-1,-1)
OpenRec(_zone_test)
```

;   This tablet zone match with the circle in the figure 151.
```
WaitForTabZoneAt(6072,2981,6248,2805,TRUE,FALSE)
CloseRec
```


Figure 150: "masque_5_mots.jpg" picture used for unmasking.

The following figure shows the picture that serve as background and mask, with the representation of an unmasking zone in a dotted line.
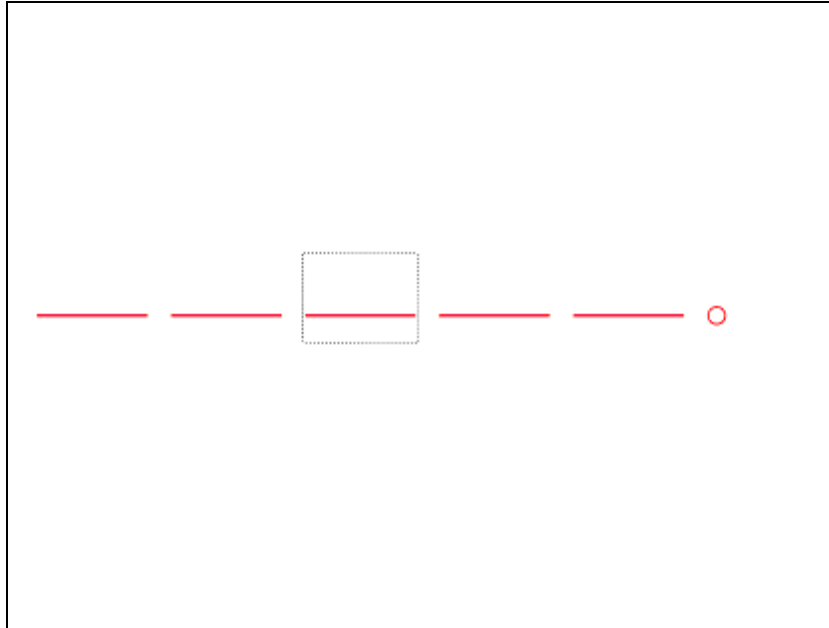
Figure 151: "fond_blanc_5 mots.jpg" (white_background_5_words).

III.4. Load a list and show its content

The following script loads a list of item from a file (the file named V2_liste.txt, is found in the *Stimuli* folder), then displays it's content, item by item.

```
LoadList(v2_liste.txt)
WaitForKeypressMsg(%M% items in the list,-1,-1,FALSE)

:Start
  WaitForKeypressMsg(Item %I%:%L%,-1,-1,FALSE)
  JumpToIfNumberIs(StopIt,%M%,FALSE)
  JumpTo(Start,FALSE)

:StopIt
```

Using %M% (number of items in the list) allows you to have all the items in the whole list displayed, however many there are.
`WaitForKeypressMsg(Item %I%:%L%,-1,-1,FALSE)` allows you to display the number (%I%) and content (%L%) of each item in the list.
Note: to display the picture whose name is listed in the list (at the current position), use
`DisplayPic(%L%,-1,-1,-1).`

The content of the file named "`v2_liste.txt`" is:

```
Ent1.jpg
Ent2.jpg
Ent3.jpg
pilori.jpg
studio.jpg
defile.jpg
detenu.jpg
dragon.jpg
```

The JPG pictures listed here should be found in the *Stimuli* folder.

III.5. Randomize sub-blocks of a list

The following script fills a list with 16 numbers (from 1 to 16) and randomizes two independent sub-blocks. Next, the content of the list is dipslayed, item by item.

; *Fill the list with 16 numbers, from 1 to 16*

```
:Fill_In
  AddToList(%I%)
  JumpToIfNumberIs(Next,16,FALSE)
  JumpTo(Fill_In,FALSE)
:Next
```

; *randomize two independent sub-blocks. Items 1 and 2 are not involved.*

```
RandomizeListRange(3,8)
RandomizeListRange(9,16)
```

; *Display list's items*

```
:Loop
  WaitForKeyPressMsg(Item %I%: %L%,-1,-1,FALSE)
  JumpToIfNumberIs(Stop,16,FALSE)
  JumpTo(Loop,FALSE)
:Stop
```

Note: the sequence "`JumpToIfNumberIs(Next,16,FALSE)`" then "`JumpTo(Fill_In,FALSE)`" (lines 3 and 4 of the sample above) has the same function as the command "`LoopIfLabelIsBelow(Fill_In,16,FALSE)`": if the value of the "`Fill_In`" label's counter is below 16, then jump to this label.

III.6. Use two label counters with one shifted

The following script shows how it is possible to count, starting with a value greater than 1.

```
SetLabelCounter(Start,2)
WaitForKeyPressMsg(Starting value: %I:Start%,-1,-1,FALSE)

:Start
  :Iteration
  WaitForKeyPressMsg(Iteration %I:Iteration% : %I:Start%,-1,-1,TRUE)
  LoopIfLabelIsBelow(Start,6,FALSE)
```


III.7. Have a list of words copied, with NetSync

The following script allows you to have a list of items copied by a group of participant. The trials are managed with NetSync (see Netsync, p. 104).

```
DisplayMsg(Welcome !,2000,-1,-1,FALSE)
DisplayMsg(Please, wait a moment...,-1,-1,-1,FALSE)
```

; *Wait for the start signal sent by the Master host*

```
WaitForNetSync
HideMessage
ResetList
```

```
;    Add items to the list, one by one
AddToList(pilori)
AddToList(studio)
AddToList(defile)
AddToList(detenu)

:start

   ;    Display a star and wait for a "pen press" in a zone located on the left side of the tablet
   DisplayPic(croix.bmp,-1,-1,-1)
   WaitForTabZoneAt(1420,4900,4800,26530,FALSE,FALSE)

   ;    Record the production in a file named <participant>_<item>.TAB
   openrec(_%L%)

   ;    Wait for 500 milliseconds
   WaitFor(500)

   ;    Send the number of item in the list (ex. "Item5/23") to the Master host
   SendMessageToNetSync(Item%I%/%M%)

   ;    Display the item whose number in list match the "start" label's counter value
   ;    (item n°1 at first "round" - pilori, item n°2 for the second round -
   ;    studio, etc.)
 ;    Hint: the extension ".jpg » is appended to the item name to call a matching picture
   DisplayPic(%L%.jpg,-1,-1,-1)

   ;     When the pen is pressed in the writing zone, clear the display
   WaitForTabZoneAt(4800,0,54200,31800,FALSE,FALSE)
   HidePicture

   ;    Wait for a "pen press" in the "end" zone. Participant's writing is displayed on screen.
   WaitForTabZoneAt(44930,23450,54200,31800,FALSE,TRUE)
   CloseRec

 ;    If the current item is the last one, jump to the "end » label
   JumpToIfNumberIs(end,%M%,FALSE)
   JumpTo(start,FALSE)
:end
DisplayMsg(Thanks for coming !,1000,-1,-1,FALSE)
```

<u>Note</u> : to "synchronize" the copy task item by item, i.e. to have the participants to wait before each trial (a collective "Go" is sent par the Master host to "unlock" the trial), the command `WaitForNetSync` could have been inserted at the beginning of the "start" block, after a message saying to wait, like in the following example:

```
:start

DisplayMsg(Be ready...,-1,-1,-1,FALSE)
WaitForNetSync
HideMessage

   ;    Display a cross...
   DisplayPic(croix.bmp,-1,-1,-1)
```

III.8. Dictation of a text

The following scriptallows you to dictate a text, previously recorded as an audio file (Wave format) and save into the *Stimuli* folder.

;   Display the instructions contained in the text file named "Instructions.txt".
```
DisplayText(Intsructions.txt,-1)
```

;   Wait for the participant to press the pen in the zone labeled "Start dictation".
;   Parameter: wait until the press is lifted up to go on
```
WaitForTabZoneAt(25670,20780,29400,18815,FALSE,TRUE)
```

;   Clear the instructions text
```
HideText
```

;   Open the acquisition file<participant>_dictation.TAB
```
OpenRec(_dictation)
```

;   Play the audio file and go on with the script without waiting
```
PlaySound(dictation.wav,FALSE)
```

;   Wait for the pen to be pressed in the zone labeled « I'm finished » on the tablet.
;   The participant's writing is displayed until this event.
```
WaitForTabZoneAt(25970,5565,28900,3915,TRUE,FALSE)
```

;   Close the acquisition file.
```
CloseRec
```

III.9. Copying along with a mental load

The following script allows you to have a participant to copy a word after having heard a list of numbers. Next, he'll be to recall (write down) these numbers.

```
WaitForKeyPressPic(wait.bmp,-1,-1)
```

;   Plays the audio recording of the numbers 1-3-8-9-5.
```
Playsound(13895.wav,TRUE)
DisplayPic(croix.bmp,5000,-1,-1)
```

;   Open the acquisition file named <participant>_5Ejournal.TAB for the copy.
```
OpenRec(_5Ejournal)
```

;   Display the word to copy
```
DisplayPic(E-journal.bmp,-1,-1,-1)
```

;   Write until the pen is presses in the tablet "end" zone
```
WaitForTabZoneAt(6580,3830,7100,2270,TRUE,TRUE)
HidePicture
```

;   Close the acquisition file and open a new acquisition file for the numbers transcription
```
CloseRec
OpenRec(_5EjournalCH)
```

;   Display the start signal to write numbers down.
```
DisplayPic(B-rappelchif.bmp,-1,-1,-1)
```

;   Write until the pen is pressed in the tablet "end" zone.
```
WaitForTabZoneAt(6580,1500,7100,0,TRUE,TRUE)
CloseRec
HidePicture
```

V.   ACQUISITION SESSION LOG FILE (SCRIPT)

A file is generated for each acquisition session (*<Data Directory><ParticipantID>*.LOG). This contains a trace of the main events of interaction with the participant (displays, recordings, participant's answers, etc.), with the corresponding time (in milliseconds) of the event. The start of timing (the zero hour) coincides with the beginning of the acquisition session (when the "Go" button is pressed).
Each line in the LOG file begins with *<Time>* followed by the command and it's parameters. In the following list, the parameters between **<>** are replaced (in a true log file) with their value (see "script commands", p. 63, for name and parameter values).

WaitForTabZoneAt *<X1> <Y1> <X2> <Y2> <CanDraw> <MustLeave>*

WaitForTabZones *<CanDraw> <MustLeave>*

ZoneSelected *<selected zone label>*

RecStandard *<participant file> <UseBack> <PictureFileName> <UseIm1> <PictureFileName> <zone1 X1> <Y1> <X2> <Y2> <ShowOnStart> <HideOnPress> <UseIm2> <PictureFileName> <zone2 X1> <Y1> <X2> <Y2> <zone Fin X1> <Y1> <X2> <Y2>*

Pictures shown and hidden during Rec_Standard (trigger zones) are recorded in LOG:
- Show Image<numero>
- Hide Image<Numero>

OpenRec *<participant file>*

CloseRec

DisplayMsg *<Message> <X> <Y>*

MessageHidden

DisplayText *<TextFileName>*

TextHidden

DisplayPic *<PictureFileName> <X> <Y>*

PictureHidden

DisplayImageList *<ListFileName> <X> <Y> <DurationPerPicture>*

DisplayAVI *<VideoFileName> <X> <Y>*

AVIEnd

WaitForKeyPress

KeyPressed *<touché appuyée>*

WaitFor *<Durée>*

PlaySound *<WavFfileName> <Wait>*

SoundEnd

WaitForNetSync

SendMessageToNetSync *<message>*

TabletMasking *<TabMaskFile> <TabMaskMode> <ModeParam>*

TabletTimeShift *<Delay>*

V.   MENU TREE VIEW

File   Script editor
            Acquisition   ▶   Simple
                                    Script
                                    NetSync

            New analysis
            Reopen
            Session
            Imports   ▶   G-Studio
            Save
            Save as
            Exports   ▶   Text file
                                    G-Studio
            Close
            Configuration   ▶   Acquisition
                                    Analysis
                                    Open
                                    Save as
                                    Save as default
                                    Reload defaults
            Quit

Tests   Tablet   ▶   Visual
                               Performance
            Eye tracker   ▶   Visual
                               Performance

Protocol   Thresholds
               History
               Description
               Preview   ▶   Circle   ▶   Pauses
                                                Fixations
                                                AOI scanpath
                             Temperature ▶   Pauses
                                                Pauses by duration classes
                                                Fixations
                             Graph   ▶   XY coordinates
                                                Gaze-pen distance

               Close preview
               Sub-analysis
               Capture to picture

Tools   Cancel
            Shilft layers
            Visual zones (AOI)
            Code -1 "out-of-field" Eye data
            Correct "out-of-field" erroneous Eye data
            Build fixations (barycenter method)
            Build fixations (multi-threshold method)
            Build down pauses
            Tablet zones auto-coding
            Word separation   ▶   New
                                             Continuer
                                             Open
                                             Save as

            Reset codes
            Replace codes
            Aggregate redundant codes

	Sequences	▶	Edit sequence list
			Clear sequence list
			Open
			Save as

Extraction	Events	▶	All events
			During pauses & fixations
			Trigger zones (Simple)
	Pauses		
	During pauses	▶	Eye activity
			Fixations
			Eye activity in AOI
			Fixations in AOI
			Eye activity in AOI summary
			Fixations in AOI summary
	Fixations		
	Visual zones (AOI)	▶	Eye activity
			Fixations
			Eye activity summary
			Fixations summary
			Route as string
	Tracing dynamics		
	Preferences	▶	General
			Range
			Data
			Codes
			Tracing dynamics
			Visual zones (AOI)

Display	Toolbars	▶	Clock
			Sequences
			Retrospective
			Zoom
			Keyboard recorder
			Events
			Pauses
			Fixations
			Eye codes
			Tab codes
			Eye-Tab distance
			Customize
	Information		
	Enlarge (x2)		
	Shrink (/2)		
	Menu shortcuts		
	Refresh		

?	Help index		
	Manual		
	System informations		
	Conversions	▶	Angle -> pixels
			Tablet units <-> distance
			Tablet units <-> pixels
	About		

VI. FILENAME EXTENSIONS

This table explains the file extensions recognized or generated by Eye and Pen.

Extension	Meaning
_1	"Magneto" file (import / export)
AOI	Visual Area Of Interest file
BMP	Bitmap picture (import / export)
EMF	Image Enhanced Metafile Format (importation)
EWK	Analysis file, eye (gaze) data
EYE	"Eye tracker" acquisition data
EYE.TXT	"Raw" eye data export file (text-only file)
GIF	Image Graphics Interchange Format (importation)
HST	History file for a protocol (record of data reduction, etc.)
ICO	Icon picture (import)
INI	Eye and Pen configuration file
JPG/JPEG	Jpeg picture(import / export)
LOG	Record of main interactive events during an acquisition session (Script)
RTR	Retrospective comments file for a protocol
SEQ	Protocol sequence file
TAB	"Tablet" acquisition data
TAB.TXT	"Raw" tablet data export file (text-only file)
TWK	Analysis file, tablet data
TXT	Text file (export)
WAV	Audio file (Wave format)
WDS	Word separation data for a protocol
WMF	Windows Metafile Format picture (import)

VII. FREQUENTLY ASKED QUESTIONS (FAQ)

VII.1. Calculate a picture's position on screen

If your picture should be placed at the upper left border of the screen that's easy: coordinates are 0 and 0 (the origin of the screen frame is top and left).

To set a picture at 10cm from the left side of the window on a 1024x768 pixels screen resolution (to know the screen resolution, see the Windows control panel, Display, Parameters): I will measure (with a ruler) the display surface's horizontal width. My15 inches LCD screen measures 30,4 cm. My horizontal resolution is 1024, so 10 cm will represent: (1024 / 30,4) x 10 = 336,84, i.e.about 337 pixels. I proceed the same way for the vertical coordinate, and that's the trick !

VII.2. Other matters

You'll find this list (frequently updated) on the web site http://www.eyeandpen.net, in the FAQ section.

VIII. DATA EXTRACTION COLUMN HEADERS

The following listing explains what each column header means when extracting data from an analysis.

Num_T	Tablet event number (Tab)[4].
Xbeg_T	Horizontal coordinate at the beginning of the event
Ybeg_T	Vertical coordinate at the beginning of the event
Xend_T	Horizontal coordinate at the end of the event
Yend_T	Vertical coordinate at the end of the event
Dist_T	Distance between the beginning and end of the event
Press	Pressure exerted on the pen tip
Tbeg_T	Timestamp for the beginning of the tablet event
Tend_T	Timestamp for the end of the tablet event
Dur_T	Tablet event duration
Code_T	Code assigned to the tablet event
Cat_T	Categorization of the tablet event: ▪ Up: pause with the pen lifted up ▪ Dn: pause with the pen pressed on the tablet ▪ Mov: pen movement
Num_E	Eyetracking data event number (Eye)
Xbeg_E	Horizontal coordinate at the beginning of the event
Ybeg_E	Vertical coordinate at the beginning of the event
Xend_E	Horizontal coordinate at the end of the event
Yend_E	Vertical coordinate at the end of the event
Dist_E	Distance between the beginning and end of the event
Tbeg_E	Timestamp for the beginning of the eyetracking data event
Tend_E	Timestamp for the end of the event
Dur_E	Event duration
Code_E	Code assigned to the Eye event
Cat_E	Categorization of the event: ▪ Fix: fixation ▪ Mov: eye movement
Dist_T_E_beg	Distance between pen location and gaze location at the beginning of the event
Dist_T_E_end	Distance between pen location and gaze location at the end of the event

Sequences

SeqNum	Sequence number
Begin	Sequence start time
End	Sequence end time
Label	Sequence label (description)

Trigger zones (Simple acquisition)

TriggerNum	Trigger zone number
"Up" pause before the pen is pressed in the zone for the first time	
PrevUpNum	Pause number (tablet event)
PrevUpXBeg	Horizontal coordinate at the beginning of the pause

[4] Tablet and eyetracking data have their own distinct numbering.

PrevUpYBeg	Vertical coordinate at the beginning of the pause
PrevUpXEnd	Horizontal coordinate at the end of the pause
PrevUpYEnd	Vertical coordinate at the end of the pause
PrevUpXYDist	Distance between the beginning and end of the pause
PrevUpStartTime	Timestamp for the beginning of the pause
PrevUpEndTime	Timestamp for the end of the pause
PrevUpDuration	Pause duration
PrevUpCode	Code assigned to the pause event

1st event (sample) when the pen is pressed in the zone

NumBeg_T	First tablet event number (Tab)
XBeg_T	Horizontal coordinate of the event
YBeg_T	Vertical coordinate of the event
Tbeg_T	Timestamp of the event
CodeBeg_T	Code assigned to the event
EffectPressDur	Actual duration of the pen press in the zone (excluding "pen up" durations between two successive presses in the zone), before the pen leaves the zone (pressed somewhere else outside)

Last event before the pen leaves the trigger zone

NumEnd_T	Last tablet event number (Tab)
XEnd_T	Horizontal coordinate of the event
YEnd_T	Vertical coordinate of the event
Tend_T	Timestamp of the event
CodeEnd_T	Code assigned to the event

"Up" pause following the last press of the pen in the zone (ending up outside the zone)

LastUpNum	Pause number (tablet event)
LastUpXBeg	Horizontal coordinate at the beginning of the pause
LastUpYBeg	Vertical coordinate at the beginning of the pause
LastUpXEnd	Horizontal coordinate at the end of the pause
LastUpYEnd	Vertical coordinate at the end of the pause
LastUpXYDist	Distance between the beginning and end of the pause
LastUpStartTime	Timestamp for the beginning of the pause
LastUpEndTime	Timestamp for the end of the pause
LastUpDuration	Pause duration
LastUpCode	Code assigned to the pause event

Tracing dynamics

d_TE_beg	Distance between pen location and gaze location at the beginning of the segment
d_TE_end	Distance between pen location and gaze location at the end of the event
nb_pts	Number of tracing (writing) points (events) included in the segment
distance	Summed distance between these points
duration	Summed duration between these points
speed	Mean tracing (writing) speed for the segment
Pressure	Mean pressure for the segment
nb_Excl_pauses	Number of pauses excluded from calculations
Excl_pauses_dist	Summed distance (between pen "Up" and pen "Down") between pauses excluded from calculations
Excl_pauses_dur	Summed duration of pauses excluded from calculations

Visual zones (AOIs)

nEvtTab	Number of the tablet event (during a pause)
Code_T	Code assigned to the tablet event
zone	Zone number
Code	Code assigned to the zone
Tbeg	Timestamp for the beginning
nb_pts	Number of eye events in the zone
distance	Length of gaze movements in the zone
duration	Duration of fixations within the zone
speed	Mean speed of gaze movements

IX. STRUCTURE OF "EYE AND PEN" FILES

Every file fulfilling these prescriptions can be opened within Eye and Pen 2.
The prototypes are given in Pascal 32 bits (Delphi 7). "Word" represents a 16-bit unsigned numeric type, Integer represents a 32-bit signed numeric type.

TAB FILE HEADER: (128 bytes)

Field	Type	Description
Id	Word = 137	TAB file identifier
Version	Word = 112	Current version
HeaderSize	Word	Header size (in bytes)
Dx, Dy	Word	Display X and Y sizes (e.g. 1024*768)
X1, Y1, X2, Y2	Word	Definition of the tablet coordinates
LgmmX, LgmmY	Word	X and Y tablet resolution (e.g. 200 lines /mm)
SamplingRate	Word	Acquisition frequency (e.g. 200 Hz).
CartesianOrigin	Word	(0/1). Is the coordinates system following a Cartesian orientation? (1=yes)
DecX, DecY	Integer	Data shift
MeasureUnit	Word	Measurement unit
PressureLevels	Word	Pressure levels range
Reserved	Array[0..44] of Word	Reserved for future use

Measurement unit:
- 0 : unknown
- 1 : inches
- 2 : centimeters
- 3 : degrees
- 4 : radians
- 99 : pixels

TAB DATA

Pressure	Word	Pen pressure (0 to 1023)
X, Y	Word	Pen X and Y coordinates
Time	Integer	Time in milliseconds

EYE FILE HEADER (128 bytes)

Id	Word = 149	EYE file identifier
Version	Word = 111	Current version
HeaderSize	Word	Header size (in bytes)
OcModel	Word	Eye tracker model number
EyeDegX, eyeDegY	Word	For angular data, number of degrees between two calibration points, on X and Y axes
nPtX, nPtY	Word	For angular data, number of X and Y calibration points
X1,Y1,X2,Y2	Word	Calibration coordinate system for non-angular data (e.g. 0,0,1024,768)
SamplingRate	Word	Acquisition frequency (e.g. 500 Hz)
CartesianOrigin	Word	(0/1). Is the coordinate system following a Cartesian orientation? (1=yes). Fixed to 1, yet
DecX, DecY	Integer	Data shift
Reserved	Array[0..45] of Word	Reserved for future use

EYE DATA

X, Y	Integer	X and Y coordinates
Time	Integer	Time in milliseconds

Eye tracker model number:
- 0: EyePuter
- 1: EyeLink
- 2: ASL504
- 3: iViewX
- 4 : Tablet tracking

X. REGISTRY KEYS (WINDOWS)

When you tick the option "associate Eye & Pen with .TAB and .TWK files", registry keys are added to the Windows registry. These are removed when you untick the option.

Key: HKEY_CLASSES_ROOT	**Value** (Reg_SZ)
\.tab	Ep.Data
\.twk	Ep.Analysis
\EP.Data	EP File
\EP.FileType\DefaultIcon	c:\ Program Files\Eye and Pen 2\ EP.exe,0
\EP.FileType\Shell\open\command	"c:\ Program Files\Eye and Pen 2\ EP.exe" "-F" "%1"
\EP.Analysis	EP File
\EP.FileType\DefaultIcon	c:\ Program Files\Eye and Pen 2\ EP.exe,0
\EP.FileType\Shell\open\command	"c:\ Program Files\Eye and Pen 2\ EP.exe" "-F" "%1"

XI. TROUBLESHOOTING

1- Since I plugged a second tablet in, everything has gone wrong.

WACOM tablet drivers prior to version 4.78.6 may cause problems if too many tablets are connected.
De-install the current driver (in Windows, Configuration panel, Add/Remove programs), then install the latest version.

2- Eye and Pen tells me "Wintab32.dll not found", but my tablet still seems to work in Windows.

Wintab32.dll is an interface library that is needed for Eye and Pen to communicate with your tablet's driver.
Usually, this file is installed when you install the tablet's software provided by its manufacturer. When in doubt, reinstall the driver.

3- I move the pen on the tablet (Device Test/Tablet/Visual menus), but nothing happens...

Check if the pen you are using is compatible with the tablet. For example, Wacom Intuos2 pencils are not compatible with other tablets from the same manufacturer. Also check if the tablet's driver is correctly installed and configured. Check if you have selected the tablet you actually want to use (*File/Configuration/Acquisition/Tablet* menu)

4- When I click on the *?/Manual* menu nothing is displayed

Either the EPManuel_EN.pdf file is not in the *Documents* subdirectory of Eye and Pen, or you do not own the software required to read this document (Adobe Reader®), which you can download for free from the Website http://www.adobe.fr/products/acrobat/)

XII. ERROR MESSAGES

XII.1. Eye and Pen

General

`<filename>: analysis data not found`
Does the analysis file still exist in the same directory?

`Save failed for <filename>`
Is there enough memory left? Does the same filename exist in this directory with a "read-only" attribute?

`<filename>: failed to write`
Idem as above

`<filename>: data not found`
One of the two analysis files is missing (TWK or EWK).

`not enough video memory`
The video card does not have enough video memory capacity to handle the display with the width and height you asked for.

configuration

Configuration file not found: `<filename>`

The configuration file declared in Eye and Pen is missing

The program must abort because of a missing file.

A file required by Eye and Pen is not foundt. Check your installation

Invalid picture file

The selected picture could not be read by Eye and Pen.

Recording devices already in use

You are attempting to use the tablet when it is already being used.

Save failed for `<filename>`

Either the file is in "read-only" format or your disk is full.

Incorrect numeric value! (e.g. 4,7)

You typed something that is not a valid numerical value.

Incorrect values in `<configuration filename>`

Some mandatory parameters of the configuration file either do not exist or their value is not a number.

Script

Cannot load `<picture filename>`

A picture could not be loaded. Either the name is incorrect (the file cannot be found where it is supposed to be) or there is not enough video memory available for a picture of this size.

Error line `<number>` in script `<script filename>`

The script contains a mistake. The line number reported does not include empty or comment lines.

Duplicate Label: "`<label>`"

You have created the same label twice, which is not allowed in a script.

Unknown command: "`<command>`"

This command is not known to Eye and Pen. See the command list (maybe a misspelling?).

Error in file list `<list name>`

A picture mentioned in the picture list cannot be found.

No Wintab32 tablet driver available. Abort recording.

Eye and Pen could not find the tablet driver (see IX. Troubleshooting).

No tablet connected. Abort recording.

A tablet is required...

Label not found: "`<label>`"

A command refers to a label which does not exist in the script.

Cannot find `<directory or file>`

The script mentions a directory or a file that Eye and Pen cannot find. Typo?

Failed to create folder

The NetSync client couldn't create the sub-folder for the acquisition data: is the support full or write locked ?
This message also exists for the Master host.

Authentication failed
   The connection between the Master and the NetSync client failed: the client is not recognized as valid.

No audio device detected
   Is your PC equipped with a sound card or onboard sound chip? Is it properly installed and configured in Windows?

Failed to handle audio mixer
   Eye and Pen cannot communicate with the audio device management system.

Audio mixer open failed
   Ditto.

No MCI device opened
   Probably two successive `PlaySound` commands. The system needs a little time to close the audio device before being able to re-open it with another file. Instead of several successive audio files, it is advisable to create a single file containing all these successive samples (e.g. create a dictated number series).

Eyelink

Eyelink connection lost
   A communication breakdown has occurred. Check the Ethernet link.

Can't unload Eyelink library
   When closing communication with the eye tracker, Eye and Pen unloads a specific Eyelink library. To avoid problems, it is best to restart your computer.

Failed to close EyeLink connection
   Eye and Pen could not politely close its conversation with the eye tracker.

ERROR: Start recording failed!
   Eyelink failed to start the data recording and data transmission to Eye and Pen.

ERROR: Eyelink is not connected!
   Check connections and the eye tracker network configuration in Windows.

ERROR: No link data samples received!
   A communication problem with the eye tracker. Check the configuration on the Eyelink host. Has the recording mode been engaged?

EYELINK_EXPTKIT20.DLL not loaded
   The interface library for the eye tracker could not be loaded.

EYELINK_EXPTKIT20.DLL: function "*<something>*" not supported
   Eye and Pen requested a function that is not supported by the eye tracker interface library. Have you installed the latest Eyelink library version?

FATAL: Eyelink "open connection" failed.
   Eye and Pen cannot start to "speak" with the eye tracker. Possibly a library version problem.

Cannot get window to calibrate
   Probably a system overload problem. It is safe to restart your computer.

Cannot know which eye is used
   Eyelink has returned an incorrect item of information.

EyePuter & ASL 504

Failed to get state: *<COM port>*
Eye and Pen cannot check the state of the *<COM port>* serial port through which it is supposed to communicate with the eye tracker. It is safe to restart your computer.

Failed to get timeouts for *<COM port>*
Eye and Pen could not read the timeouts values for the serial port. It is safe to restart your computer.

Failed to set *<COM port>*
Eye and Pen could not set the parameters for the <COMport> serial port. It is safe to restart your computer.

Failed to set timeouts for *<COM port>*
Eye and Pen could not set the timeouts values for the serial port.

Failed to get sampling rate
Eye and Pen should read the Eyeputer sampling rate (60, 240 or 480 Hz) before launching the acquisition, because the data format is different for each value.

Opening port failed: *<COM port>*
Eye and Pen could not open the serial port to communicate with the eye tracker. It is safe to restart your computer.

Failed to close *<COM port>*
Eye and Pen couldn't close the serial port communication channel. If you need to re-launch an acquisition, restart the computer beforehand.

No data on *<COM port>*
Eye and Pen does not receive data from the eye tracker via the serial port. Check the serial connection and the eye tracker configuration on its host system.

Edition

Only numbers allowed (range separator ".." or ",")
You typed a forbidden character in the list of codes.

Dongle   (Complete code list in Appendix XI.2. SafeNet Sentinel).

Interface not found (UX32W.DLL)
The dongle interface library (UX32W.DLL) file is missing. It must be located in the Eye and Pen program directory.

Initialize failed
The activation of the dongle management system failed.

Release License failed
A problem occurred when "closing" the dongle.

SetContactServer failed
A major problem with the dongle management system initialization.

XII.2. « SafeNet Sentinel ».

Code	Message	Code	Message
1	INVALID FUNCTION CODE	51	INVALID PORT FLAGS
2	INVALID PACKET	52	INVALID LOG PORT CFG
3	UNIT NOT FOUND	53	INVALID OS TYPE
4	ACCESS DENIED	54	INVALID LOG PORT NUM
5	INVALID MEMORY ADDRESS	56	INVALID ROUTER FLGS
6	INVALID ACCESS CODE	57	INIT NOT CALLED
7	PORT IS BUSY	58	DRVR TYPE NOT SUPPORTED
8	WRITE NOT READY	59	FAIL ON DRIVER COMM
9	NO PORT FOUND	60	SERVER PROBABLY NOT UP
10	ALREADY ZERO	61	UNKNOWN HOST
11	DRIVER OPEN ERROR	62	SENDTO FAILED
12	DRIVER NOT INSTALLED	63	SOCKET CREATION FAILED
13	IO COMMUNICATIONS ERROR	64	NORESOURCES
15	PACKET TOO SMALL	65	BROADCAST NOT SUPPORTED
16	INVALID PARAMETER	66	BAD SERVER MESSAGE
17	MEM ACCESS ERROR	67	NO SERVER RUNNING
18	VERSION NOT SUPPORTED	68	NO NETWORK
19	OS NOT SUPPORTED	69	NO SERVER RESPONSE
20	QUERY TOO LONG	70	NO LICENSE AVAILABLE
21	INVALID COMMAND	71	INVALID LICENSE
29	MEM ALIGNMENT ERROR	72	INVALID OPERATION
30	DRIVER IS BUSY	73	BUFFER TOO SMALL
31	PORT ALLOCATION FAILURE	74	INTERNAL ERROR
32	PORT RELEASE FAILURE	75	PACKET ALREADY INITIALIZED
39	ACQUIRE PORT TIMEOUT	76	PROTOCOL NOT INSTALLED
42	SIGNAL NOT SUPPORTED	101	NO LEASE FEATURE
44	UNKNOWN MACHINE	102	LEASE EXPIRED
45	SYS API ERROR	103	COUNTER LIMIT REACHED
46	UNIT IS BUSY	104	NO DIGITAL SIGNATURE
47	INVALID PORT TYPE	105	SYS FILE CORRUPTED
48	INVALID MACH TYPE	106	STRING BUFFER TOO LONG
49	INVALID IRQ MASK		
50	INVALID CONT METHOD		

XIII. WINTAB32-COMPLIANT HARDWARE MANUFACTURERS

Wintab32 is an industrial standard intended to ensure straightforward communication between computers and digitizing tools, such as tablets. This list is not exhaustive.

Company	Web site
ACECAD Enterprise Co. Ltd.	http://www.acecad.com.tw
Aiptek Inc.	http://www.aiptek.com.tw
ALTEK Corp	http://www.altek.com
Aristo Graphic Systeme GmbH & Co. KG	http://www.aristo.de
Communications Intelligence Corp.	http://www.cic.com/
Graphtec Corp.	http://www.graphteccorp.com
GTCO Calcomp Corp.	http://www.gtcocalcomp.com
IQ Automation	http://www.iq-automation.de/
Hitachi Digital Graphics	http://www.hitachi-soft.com/icg/products
KYE Systems Corp.	http://www.geniusnetusa.com/
Mutoh America, Inc.	http://www.mutoh.be/
NEC Corp.	http://www.nec.com/
Numonics Corp.,	http://www.numonics.com
Oce Graphics SA.	http://www.oce.com
Pinnacle Technologies	http://www.pinnacle.com.ph/
Seiko Instruments Inc.,	http://www.sii.co.jp/corp/eg/index.html
Sony Corp.	http://www.sony.net/
Topaz Systems Inc.	http://www.topazsystems.com/
Twinhead Corp.	http://www.twinhead.com/
WACOM Corp.	http://www.wacom.com

XIV. BIBLIOGRAPHY

Eye and Pen: methods and early results

[An updated list is held on the web site http://www.eyeandpen.net, in Publications section]

Alamargot, D., Plane, S., Lambert, E. & Chesnet, D. (2010). Using Eye and Pen Movements to Trace the Development of Writing Expertise: Case Studies of a Seventh, Ninth and Twelfth Grader, Graduate Student, and Professional Writer. *Reading and Writing,* 23, 853-888.

Alamargot, D., Dansac, C., Chesnet, D., & Fayol, M. (2007). Parallel processing before and after pauses: A combined analysis of graphomotor and eye movements during procedural text production. In M. Torrance, D. Galbraith & L. v. Waes (Eds.), *Cognitive Factors in Writing*. Dordrecht-Boston-London: Elsevier Sciences Publishers.

Alamargot, D., Chesnet, D., Dansac, C., & Ros, C. (2006). Eye and Pen: a new device to study the reading during writing. *Behavior Research Methods, Instruments and Computers,* **38**(2), 287-299.

Alamargot, D., Dansac, C., Ros, C., & Chuy, M. (2005). Rédiger un texte procédural à partir de sources : relations entre l'empan de mémoire de travail et l'activité oculaire du scripteur. In D. Alamargot, P. Terrier & J. M. Cellier (Eds.), *Production, compréhension et usages des écrits techniques au travail* (pp. 51-68). Toulouse: Octarès.

Caporossi, G., Alamargot, D., & Chesnet, D. (2004). Using the computer to study the dynamics of handwriting processes. *Lecture Notes in Computer Science, 3245*, 242-254.

Chesnet, D., & Alamargot, D. (2005). Analyses en temps réel des activités oculaires et graphomotrices du scripteur: intérêt du dispositif 'Eye and Pen'. *L'Année Psychologique, 105*(3), 477-520.

Some publications on eye movements

Inhoff, A. W., & Gordon, A. M. (1998). Eye movements and eye-hand coordination during typing. *Current Directions in Psychological Science, 6*(6), 153-157.

Inhoff, A. W., & Radach, R. (1998). Definition and computation of oculomotor measures in the study of cognitive processes. In G. Underwood (Ed.), *Eye Guidance in Reading and Scene Perception*. North Holland: Elsevier Sciences Publishers.

Inhoff, A. W., Topolski, R., & Wang, J. (1992). Saccade programming during short duration fixations: An examination of copytyping, letter detection, and reading. *Acta Psychologica, 81*(1), 1-21.

Kennedy, A., Radach, R., Heller, D., & Pynte, J. (Eds.). (2000). *Reading as a Perceptual Process*: North Holland - Elsevier Science Publishers.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124*(3), 372-422.

Reichle, E. D., Rayner, K., & Pollatsek, A. (2003). The E-Z reader model of eye-movement control in reading: comparisons to other models. *Behavioral and Brain Sciences, 26*, 445-526.

Salvucci, D. D., & Anderson, J. R. (2001). Automated eye-movement protocol analysis. *Human Computer Interaction, 16*(1), 39-86.

Salvucci, D. D. & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Eye Tracking Research and Applications Symposium* (pp. 71-78). New York: ACM Press.

Some publications on handwritten production and pauses

Alamargot, D., & Chanquoy, L. (2001). *Through the Models of Writing*. Dordrecht-Boston-London: Kluwer Academic Publishers

Chanquoy, L., Foulin, J.-N., & Fayol, M. (1990). Temporal management of short text writing by children and adults. *Cahiers de Psychologie Cognitive, 10*(5), 513-540.

Chesnet, D., Guillabert, F., & Espéret, E. (1994). G-Studio: un logiciel pour l'étude en temps réel des parameters temporels de la production écrite. *L'Année Psychologique, 94*, 115-125.

Foulin, J.-N. (1995). Pauses et débits: les indicateurs temporels de la production écrite. *L'Année Psychologique, 95*(3), 483-504.

Foulin, J.-N. (1998). To what extent does pause location predict pause duration in adults' and children's writing. *Cahiers de Psychologie Cognitive, 17*(3), 601-620.

Levy, C. M., & Ransdell, S. (Eds.). (1996). *The Science of Writing : Theories, Methods, Individual Differences, and Applications*. Mahwah: Laurence Erlbaum Associates

Matsuhashi, A. (1981). Pausing and planning: The tempo of written discourse production. *Research in the Teaching of English, 15*(2), 113-134.

Schilperoord, J. (1996a). The distribution of pause time in written text production. In G. Rijlaarsdam, H. van den Bergh & M. Couzijn (Eds.), *Theories, Models and Methodology in Writing Research* (pp. 21-35). Amsterdam: Amsterdam University Press.

Schilperoord, J. (1996b). *It's About Time. Temporal Aspects of Cognitive Processes in Text Production*. Amsterdam - Atlanta, Ga.: Rodopi.

Schilperoord, J., & Sanders, T. (1999). How hierarchical text structure affects retrieval processes: Implications of pause and text analysis. In M. Torrance & D. Galbraith (Eds.), *Knowing What to Write: Conceptual Processes in Text Production* (pp. 13-30). Amsterdam: Amsterdam University Press.