

## Table Of Contents

Table Of Contents .....	1
Getting Started .....	2
Step 1 - Raw data conversion .....	2
Step 2 – Event code extraction .....	3
Step 3 – EEG and event display .....	4
Step 4 – Artifact rejection and response averaging .....	4
Step 5 – Evoked response display and mapping .....	5
Step 6 – Time-frequency analysis of single trials .....	5
Step 7 – Time-frequency display .....	6
CRNL .....	6



Published on *elan* (<http://elan.lyon.inserm.fr>)

[Home](#) > [Printer-friendly PDF](#) > [Printer-friendly PDF](#)

## Getting Started

ELAN consists in several stand-alone programs that can be run in a Linux shell by typing a command line. Two types of program can be found:

- computation programs  
usually consisting in a command line followed by arguments: these programs perform several types of computation on the raw continuous signals, or on averaged responses; they allow classical average response analysis, or analysis in the time-frequency domain.
- graphic viewing  
programs allowing to interactively view the different types of data : continuous eeg, averaged responses, and time-frequency representation. All three programs are highly user-friendly, and allow a comprehensive interactive viewing of all aspects of the data, including topographical mapping and measurements.

The first step consists in converting your own continuous EEG or MEG data files to the ELAN internal format. Thereafter, all processing will only concern files in ELAN format (continuous data files [**.eeg** files], average responses [**.p** files], trigger position [**.pos** files], time-frequency data [**.tf** files], ...). Here below is a more detailed description of the different steps to achieve, from continuous raw data to ERP or time-frequency viewing:

- [Step 1: Raw data conversion](#) <sup>[1]</sup>
- [Step 2: Event code extraction](#) <sup>[2]</sup>
- [Step 3: EEG and event display](#) <sup>[3]</sup> (optional)
- [Step 4: Artifact rejection and response averaging](#) <sup>[4]</sup>
- [Step 5: Evoked response display and mapping](#) <sup>[5]</sup>
- [Step 6: Time-frequency analysis of single trials](#) <sup>[6]</sup>
- [Step 7: Time-frequency display](#) <sup>[7]</sup>

AJOUTER ICI IMAGE ELAN OVERVIEW

## Step 1 - Raw data conversion

Raw signals recorded in continuous or epoch mode have to be converted to the ELAN **.eeg** file format. These data files should contain signal data and information related to stimulus or response triggers. Several converters are provided for different recording systems: [neuroscan2eeg](#) <sup>[8]</sup>, [instep2epeg](#) <sup>[9]</sup>, [EGLraw2eeg](#) <sup>[10]</sup>, [micromed2eeg](#) <sup>[11]</sup>, [brainamp2eeg](#) <sup>[12]</sup>, [biosemi2eeg](#) <sup>[13]</sup>, [ctf2eeg](#) <sup>[14]</sup>, [neuromag2eeg](#) <sup>[15]</sup> ...

This conversion will create 2 files with the following extension:

**.eeg** : contains the digitized continuous signals stored by time-sample and by channel as 16-bit signed integer.

**.eeg.ent** : is a text header file that should be always associated with the above **.eeg** file. It contains information such as sampling period, number of channels, electrode names, ...

In general, all these information are stored in the original data file created by your acquisition system. However, electrode names and positions are sometimes ambiguously defined and that can corrupt further mapping. The conversion program may thus require a file with the list of electrodes used in your own montage and their coordinates according to the [electrode definition system used in ELAN](#) <sup>[16]</sup>.

In some cases (e.g. [brainamp2eeg](#) <sup>[12]</sup>, [biosemi2eeg](#) <sup>[13]</sup>, [ctf2eeg](#) <sup>[14]</sup>, [neuromag2eeg](#) <sup>[15]</sup>), the conversion program directly creates this file from the original data file, or loads a pre-defined file containing the electrode/sensor names and coordinates in the ELAN system.

For other electrode montages, the user has to create a text file containing the list of electrodes (ordered as in the original data file) in which each line has the following structure:

electrode name '.' electrode number in the ELAN system (see [electrode definition in ELAN \(elec.dat\)](#) <sup>[16]</sup>)

Here is an example for a 32 electrode montage (31 EEG channels and 1 EOG channel): file *montage32.elec*

*electrodes* (label)

F3z.71 (electrode name and electrode number separated by '.', according to **elec.dat**)

F4z.72  
 F7.13  
 F3.14  
 Fz.15  
 F4.16  
 F8.17  
 FT3.48  
 FC1.40  
 FC2.41  
 FT4.49  
 T3.8  
 C3.9  
 Cz.10  
 C4.11  
 T4.12  
 TP3.50  
 CP1.42  
 CP2.43  
 TP4.51  
 Ma1.85  
 T5.3  
 P3.4  
 Pz.5  
 P4.6  
 T6.7  
 Ma2.86  
 O1.1  
 PP13.121  
 PP24.122  
 O2.2  
 EOG.119

The conversion of a Neuroscan continuous file, *mydata.cnt*, corresponding to this 32 electrode montage, is thus obtained by the following command line:

```
neuroscan2eeg [8] mydata.cnt montage32.elec myeegfile
```

with *myeegfile* being the prefix of ELAN **.eeg** files created in the current directory (*myeegfile.eeg* and *myeegfile.eeg.ent*) corresponding to the continuous signals in *mydata.cnt* in the current directory. The conversion program will use the *montage32.elec* to get the specification (name and number in **elec.dat**) of each electrode.

You may contact us for specific electrode montages.

## Step 2 – Event code extraction

The file *myeegfile.eeg* now contains the digitized signals and the event codes for stimuli and responses. For further processing, these event codes and positions in time have to be extracted by the following command line:

```
eegpos [17] myeegfile.eeg myeegfile.pos
```

This will create the text file *myeegfile.pos* containing the following information:

(time sample when the event occurred) (event code) (rejection status: see [Step 4](#) [4])

```

6745 10 0
7785 10 0
8765 10 0
9745 20 0
9995 100 0
10755 10 0
11775 10 0
.....

```

In this example (odd-ball paradigm), sampling rate was 1000 Hz, and stimuli were presented with an onset-to-onset inter-stimulus interval of about 1000 ms (randomized). Standard stimuli had trigger code 10 and target had code 20. Subjects had to press a button after target, generating an event code 100. The reaction time in this particular example for the target was 250 samples (9995-9745), i.e., 250 ms.

The event position file *myeegfile.pos* is a text file that can be easily processed for possible event recoding (see **poschseq**)

or reaction time computation (see **posrt**). It can also be transformed by user's specific software (Matlab or C program). More details in [Event Processing Tools](#) [18].

## Step 3 – EEG and event display

The raw signals can be displayed by the program **eeg** [19] (either by typing the command line **eeg** and interactively loading the **myeegfile.eeg** file, or by typing **eeg myeegfile.eeg** to directly load the file of interest). **eeg** [19] is an interactive graphic program. Here are some tips for getting started with **eeg**:

- Channels to be displayed can be selected by the 'Channel Selection' button (menu bar).
- Continuous signal can be browsed by clicking the right- (forward) or left- (backward) button of the mouse.
- A cursor can be displayed by clicking the middle-button (or both right- and left mouse buttons). It can then be moved by the right and left arrows on the keyboard and provides time and amplitude measures for each channel.
- Time and amplitude scales can be modified by the 'Scale' button (menu bar) or in the right menus. Each thick vertical line corresponds to 1 second.
- [Event code file](#) [2] (**myeegfile.pos**) can be loaded by the 'Event' button (menu bar). Event codes appear in the upper and lower parts of the display by blue ticks and event code numbers. The signal is browsed by jumping from one event to the other (right, forward or left, backward mouse buttons).

## Step 4 – Artifact rejection and response averaging

Once the continuous raw signals have been converted into the ELAN **.eeg** file format, and the event codes have been extracted in the **.pos** file (via the program **eegpos** [17]), it is possible to compute the average evoked potentials (EP) of interest using the command line:

```
eegavg myeegfile.eeg myeegfile.pos myparfile.par myeegfile.rej.pos +v
```

The parameters necessary for artifact rejection and EP averaging must be specified in a **.par** texte file. These parameters include:

- the number and list of event codes defining the EPs to average,
- the numbers of time samples in the pre- and post-stimulus periods (for each event code),
- the start and stop latencies for baseline correction (for each event code),
- the list of channels to be averaged,
- the list of maximum and minimum threshold values (or other criteria) for amplitude artifact rejection (per channel)... (See [eegavg](#) [20] for more detailed information).

### STRUCTURE OF THE **.par** FILE

Each line of the **.par** text file begins with a label specifying the type of parameter followed by the user's parameters. For example, in the case of an odd-ball paradigm in which the standard stimuli had trigger code 10 and targets had code 20, the **myparfile.par** file used to compute the EPs to standards and targets (with a 100-ms prestimulus baseline correction and artifact rejection for any signal exceeding  $\pm 80 \mu\text{V}$  at any electrode) will have the following form:

(label)	(parameters)	(comments)
<i>fileprefix</i>	myepfile	prefix of the output <b>.p</b> files
<i>nb_eventcode</i>	2	nb of event codes to process
<i>list_eventcode</i>	10 20	list of event codes to process
<i>prestim_nbsample</i>	200 200	nb samples in the pre-stim period; one val/evt code
<i>poststim_nbsample</i>	800 800	nb samples in the post-stim period; one val/evt code
<i>baseline_eventcode</i>	10 20	list of the event codes used to compute the baseline
<i>baseline_msec_start</i>	-100 -100	baseline start latencies (ms); one val/evt code
<i>baseline_msec_stop</i>	0 0	baseline stop latencies (ms); one val/evt code
<i>ep_channel_flag</i>	1 1 1 1 0 0	list of the channels to average
<i>ep_channel_rejtype</i>	1 1 1 1 0 0	rejection types per channel: 0,1 or 2
<i>rej_max_val</i>	80 80 80 80 0 0	
<i>rej_min_val</i>	-80 -80 -80 -80 0 0	

This structure of the **.par** file allows to compute a set of ERPs with a great flexibility in the choice of the parameters (rejection criteria, baseline corrections...). See the complete list of labels and possible parameters of the **.par** file in [eegavg](#) [20].

The outputs of the **eegavg** program are:

- the evoked potentials (**.p**) files (one **.p** file by event code), with the names formed by the prefix specified in the **.par** file ('fileprefix' label), and extension formed by the event code,

- a new **.pos** text file (e.g. **.rej.pos** file) containing the positions of events whose the codes are specified in the **.par** file ('list\_eventcode' label), with the rejection status '0' for the non-rejected event codes, or '8' for rejected codes.

Thus in the example above, the **eegavg** program will use as input files:

**myeegfile.eeg, myeegfile.eeg.ent, myeegfile.pos, myparfile.par**

It will create as output files:

- Two **.p** files (**myepfile.10.p, myepfile.20.p**),
- A new **myeegfile.rej.pos** file containing the following information:

```
6745 10 0
7785 10 0
8765 10 8
9745 20 0
10755 10 0
11775 10 8
.....
```

This new **.pos** text file can again be easily processed for event recoding... (see [Event Processing tools](#) <sup>[18]</sup>).

## Step 5 – Evoked response display and mapping

Having computed the averaged evoked potentials (**.p** files), it is possible to display them using **erpa** <sup>[21]</sup>, a highly user-friendly graphic program. **Erpa** results from 20-years interactions between electrophysiologists and software developers, and thus covers a comprehensive and thorough viewing of all aspects of the data: time curves, topography (potential and scalp current density mapping) and measurements.

Typing the command line **erpa** will open a screen window with a general menu bar presenting 6 buttons: '**Files**', '**Curves**', '**Maps**', '**Models**', '**Transform**', '**Quit**'. Each button commands a pull-down menu allowing to interactively specify the display parameters. ('**Models**' is inoperative at the moment).

**.p** file loading.

The first step is to click the '**Files**' button in the general menu bar, then the '**Load**' button to select the **.p** files to display. It is possible to '**load**' / '**unload**' several **.p** files to process several responses on the same display (superposition of time curves, juxtaposition of topographic maps...). If two or more files do not have the same length (time samples), **erpa** can align them ('**Align**' button). It is also possible to specify the amplitude units ( $\mu$ V, fT...).

**.p** curves processing (general menu bar, '**Curves**' button).

The time curves of the **.p** files loaded can be drawn over their whole latency range or within specified latency windows (click '**Time window**'). Clicking the '**Draw curves**' button displays the **.p** curves in a new screen window, with a secondary menu bar at the top of the screen. This menu bar offers a number of options as:

- select a subset of files to draw among the initially loaded files ('**Select Files**' button)
- select display options ('**Display**' button) allowing among others to specify:
  - the time and amplitude scales ('**Scales**'),
  - a personalized presentation format ('**Curves Layout**') that can be edited, saved, or loaded, allowing to have the same personalized curve presentation for a set of data,
  - the color and type of lines for the different **.p** files ('**Curves attributes**'),
  - the file legends (by default, the legend is the **.p** file name),
  - positivity up or down
- zooming on one electrode ('**Zoom**' button): this will open a new window
- spectrum analysis ('**Tools**' button, then '**Spectrum**')
- perform various measurements ('**Measures**') at one latency (amplitudes at each channel) or between two latencies (peak amplitudes and latencies, mean amplitudes, amplitude ranges, spatial or time RMS...). The latencies are selected by one or two cursors made active/inactive by clicking the '**cursor**' button at the bottom left of the curve display. The measures can be saved in external files to be processed via standard statistic programs.

In addition, when one cursor is selected, it is possible to draw one spatial map per **.p** file at the cursor latency by clicking the '**Map**' button. This opens a new screen window, at the bottom of which the '**Display Options**' and '**Map Options**' buttons allow to select the maps parameters (voltage or SCD maps, type of views (left, right, top, back...), amplitude scales (global, per file, value), ... see more information below). Moving the latency cursor allows to dynamically follow the topographic changes of the different **.p** responses.

If two cursors are selected, it is possible to draw the spatiotemporal maps between the two latencies delimited by the cursors.

Topographic mapping (general menu bar, '**Maps**' button).

## Step 6 – Time-frequency analysis of single trials

**tfavg eegdata.eeg eegdata.rej.pos param.par 1+v**

*under construction*

## Step 7 – Time-frequency display

tfviz

*under construction*

Lyon Neuroscience Research Center - Brain Dynamic and Cognition team

CRNL



---

Source URL: [http://elan.lyon.inserm.fr/?q=man\\_getting\\_started](http://elan.lyon.inserm.fr/?q=man_getting_started)

**Links:**

- [1] <http://elan.lyon.inserm.fr/?q=node/10>
- [2] <http://elan.lyon.inserm.fr/?q=node/11>
- [3] <http://elan.lyon.inserm.fr/?q=node/12>
- [4] <http://elan.lyon.inserm.fr/?q=node/13>
- [5] <http://elan.lyon.inserm.fr/?q=node/14>
- [6] <http://elan.lyon.inserm.fr/?q=node/15>
- [7] <http://elan.lyon.inserm.fr/?q=node/16>
- [8] <http://elan.lyon.inserm.fr/?q=node/49>
- [9] <http://elan.lyon.inserm.fr/?q=node/34>
- [10] <http://elan.lyon.inserm.fr/?q=node/29>
- [11] <http://elan.lyon.inserm.fr/?q=node/32>
- [12] <http://elan.lyon.inserm.fr/?q=node/22>
- [13] <http://elan.lyon.inserm.fr/?q=node/21>
- [14] <http://elan.lyon.inserm.fr/?q=node/23>
- [15] <http://elan.lyon.inserm.fr/?q=node/62>
- [16] <http://elan.lyon.inserm.fr/?q=node/9>
- [17] <http://elan.lyon.inserm.fr/?q=node/65>
- [18] <http://elan.lyon.inserm.fr/?q=node/55>
- [19] <http://elan.lyon.inserm.fr/?q=node/36>
- [20] <http://elan.lyon.inserm.fr/?q=node/56>
- [21] <http://elan.lyon.inserm.fr/?q=node/37>