

Table Of Contents

Table Of Contents	1
Conversion Tools	2
EGlav2ep	2
EGraw2eeg	4
alphamap2eeg	5
ascii2eeg	6
biosemi2eeg	7
brainamp2eeg	8
coil2mat	10
ctf275_2coil	11
ctf2eeg	12
ctf2ep	13
ctfmrk2pos	15
ctfssv2ep	16
daq2eeg	17
edf2eeg	17
eeg2fieldtrip	18
eeg2mat	19
elan2eeglab	20
ep2fieldtrip	21
ep2mat	21
ep2nutmeg	22
epeg2spm	23
epmeg2spm	24
gdf2eeg	25
igor2eeg	25
instep2epeeg	26
mat2eeg	27
mat2ep	28
mat2tf	28
micromed2eeg	30
micromed2ep	31
neuromag2eeg	32
neuroscan2eeg	32
pos2ctfmrk	33
spike2_2eeg	34
tf2mat	35
CRNL	35



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

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

Conversion Tools

- The 4 fundamental ELAN files formats (**.eeg**, **.pos**, **.p**, **.tf**) are convertible into/from Matlab file formats ([eeg2mat](#) [1], [mat2eeg](#) [2], [ep2mat](#) [3], [mat2ep](#) [4], [tf2mat](#) [5], [mat2tf](#) [6]). Note that the **.pos** file is a text file that can be easily processed or transformed for possible event recoding via any editor or user's specific software (see [Event Processing Tools](#) [7]).
- As we have seen, most raw data acquired on commercial EEG systems are convertible through specific programs into ELAN **.eeg** file formats (see [Raw data conversion](#) [8] for more details). Several converters are provided for different recording systems: [biosemi2eeg](#) [9], [brainamp2eeg](#) [10], [ctf2eeg](#) [11], [EGraw2eeg](#) [12], [instep2epeg](#) [13], [micromed2eeg](#) [14], [neuroscan2eeg](#) [15], [neuromag2eeg](#) [16].
- In addition, **.eeg** file formats can be obtained from other file formats: Alpha-Map ([alphanmap2eeg](#) [17]), ASCII ([ascii2eeg](#) [18]), EDF ([edf2eeg](#) [19]), GDF ([gdf2eeg](#) [20]) or IGOR ([igor2eeg](#) [21]) file formats.
- MEG evoked responses from CTF can be converted into ELAN **.p** file formats ([ctf2ep](#) [22], [ctfssv2ep](#) [23]). All **.p** files can be converted in turn into Fieldtrip ([ep2fieldtrip](#) [24]), Nutmeg ([ep2nutmeg](#) [25]) or SPM ([epeg2spm](#) [26], [epmeg2spm](#) [27]) file formats.
- CTF marker files containing event codes for stimuli and responses can be converted into ELAN **.pos** files ([ctfmrk2pos](#) [28]) to be possibly processed with specific functions, and recoded back into CTF marker file formats ([pos2ctfmrk](#) [29]).

EGLav2ep

• Description

Conversion tool for EGIS (Geodesic) averaged files to evoked potebtial **.p** file format.

• Usage

EGLav2ep *myEGfile..egis myepfile myparfile.par*

with :

- *myEGfile..egis*: EGIS file to convert (with extension).
- *myepfile*: output **.ep** file (no extension).
- *myparfile.par*: text file containing conversion parameters (electrode names and numbers) (with extension).

• Fields of parameter file and example

```

electrodes
electrodes
E128_E1.5851
E128_E2.5852
E128_E3.5853
E128_E4.5854
E128_E5.5855
E128_E6.5856
E128_E7.5857
E128_E8.5858
E128_E9.5859
E128_E10.5860
E128_E11.5861
E128_E12.5862
E128_E13.5863
E128_E14.5864
E128_E15.5865
E128_E16.5866
E128_E17.5867
E128_E18.5868
E128_E19.5869
E128_E20.5870
E128_E21.5871
E128_E22.5872
E128_E23.5873
E128_E24.5874
E128_E25.5875
E128_E26.5876
E128_E27.5877
E128_E28.5878
E128_E29.5879
E128_E30.5880
E128_E31.5881
E128_E32.5882
E128_E33.5883
E128_E34.5884
E128_E35.5885
E128_E36.5886

```

E128_E37.5887
E128_E38.5888
E128_E39.5889
E128_E40.5890
E128_E41.5891
E128_E42.5892
E128_E43.5893
E128_E44.5894
E128_E45.5895
E128_E46.5896
E128_E47.5897
E128_E48.5898
E128_E49.5899
E128_E50.5900
E128_E51.5901
E128_E52.5902
E128_E53.5903
E128_E54.5904
E128_E55.5905
E128_E56.5906
E128_E57.5907
E128_E58.5908
E128_E59.5909
E128_E60.5910
E128_E61.5911
E128_E62.5912
E128_E63.5913
E128_E64.5914
E128_E65.5915
E128_E66.5916
E128_E67.5917
E128_E68.5918
E128_E69.5919
E128_E70.5920
E128_E71.5921
E128_E72.5922
E128_E73.5923
E128_E74.5924
E128_E75.5925
E128_E76.5926
E128_E77.5927
E128_E78.5928
E128_E79.5929
E128_E80.5930
E128_E81.5931
E128_E82.5932
E128_E83.5933
E128_E84.5934
E128_E85.5935
E128_E86.5936
E128_E87.5937
E128_E88.5938
E128_E89.5939
E128_E90.5940
E128_E91.5941
E128_E92.5942
E128_E93.5943
E128_E94.5944
E128_E95.5945
E128_E96.5946
E128_E97.5947
E128_E98.5948
E128_E99.5949
E128_E100.5950
E128_E101.5951
E128_E102.5952
E128_E103.5953
E128_E104.5954
E128_E105.5955
E128_E106.5956
E128_E107.5957
E128_E108.5958
E128_E109.5959
E128_E110.5960
E128_E111.5961
E128_E112.5962
E128_E113.5963
E128_E114.5964
E128_E115.5965
E128_E116.5966
E128_E117.5967
E128_E118.5968
E128_E119.5969
E128_E120.5970
E128_E121.5971
E128_E122.5972
E128_E123.5973
E128_E124.5974
E128_E125.5975
E128_E126.5976

E128_E127.5977
E128_E128.5978

Output channel name list :

Names and numbers corresponding to **elec.dat** definition for converted channels, or name and -1 if the channel is not defined in elec.dat.

The order should be the same as in the input EGIS file.

- **Example**

Conversion from EGIS with GSN HydroCell 128 channels :

`EGlav2ep "my-file-to-convert.egis" myoutput-file conv-GSN-128.par`

Read and convert my-file-to-convert.egis to myoutput-file.p with electrodes defined for GSN HydroCell 128 channels.

The parameter file is available to download here : [conv-GSN-128.par](#) ^[30]

Please ask if you are using another electrode configuration.

- **Comments**

- **Current version**

1.00 18-12-2012

- **History**

- 1.00 18-12-2012 (PEA) : 1st version.

- **Files**

\$ELANPATH/bin/EGlav2ep

- **See also**

[EGlav2eeg](#) ^[12]

Attachment	Size
conv-GSN-128.par ^[31]	1.78 KB

EGlav2eeg

- **Description**

Conversion tool for EGIS (Geodesic) files to **.eeg** format.

- **Usage**

`EGlav2eeg myEGfile myparfile myeegfile`

with :

- *myEGfile*: EGIS file to convert (with extension).
- *myparfile*: text file containing conversion parameters (electrode names and numbers).
- *myeegfile*: output **.eeg** file (no extension).

- **Fields of parameter file and example**

electrodes	
E128_E1.5851	
E128_E2.5852	
E128_E3.5853	
E128_E4.5854	
E128_E5.5855	
E128_E6.5856	
E128_E7.5857	
E128_E8.5858	
E128_E9.5859	
E128_E10.5860	
E128_E11.5861	
E128_E12.5862	
E128_E13.5863	
E128_E14.5864	
E128_E15.5865	
E128_E16.5866	
E128_E17.5867	
E128_E18.5868	
E128_E19.5869	
E128_E20.5870	
E128_E21.5871	
	Output channel name list :
	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
	The order should be the same as in the input EGIS file.

E128_E22.5872

• **Example**

Parameter file for SGN200 v1.0 and 2.0 128 channels : [sgn200.par](#) [32]

• **Comments**

• **Current version**

1.05 13-08-2007

• **History**

- 1.03 23-09-2005 (PEA) : support EGIS version 4, 5, 6, 7. Support data in microV. EGIS versions corresponds to data types :
 version : data type : file type
 2 : short int : continuous
 3 : short int : epoched
 4 : float : continuous
 5 : float : epoched
 6 : double : continuous
 7 : double : epoched
- 1.04 04-02-2006 (PEA) : bug fix for reading version 3 files.
- 1.05 13-08-2007 (PEA) : minor modification.

• **Files**

\$ELANPATH/bin/EGIraw2eeg

• **See also**

[EGlav2ep](#) [33]

Attachment **Size**

[sgn200.par](#) [32] 1.78 KB

alphamap2eeg

• **Description**

Conversion tool for Alpha-Map files to **.eeg** format.

• **Usage**

alphamap2eeg *myAlphaMapfile.map* [*myeegfile myparfile*]

with :

- *myAlphaMapfile.map* : Alpha-Map file to convert (with extension).
- *myeegfile* : output **.eeg** file (no extension).
- *myparfile* : text file containing conversion parameters.

• **Fields of parameter file and example**

channel_type_nb 4	Number of types of data defined below.
channel_type_list SPK 1 LFP 0 ANC 0 DIG 2 CH 3	Data types list : sequence of labels and codes. Labels correspond to the prefixes of the channel names used for conversion (in <i>Alpha-Map</i> file). Digital channels (DIG) are used for event creation but are not converted. The following codes are used : 0 : continuous signal channel (LFP, sEEG, ...). 1 : spike channel. These channels are converted to continuous channels and the spikes are averaged along time. 2 : trigger channel (events). 3 : digital channel to convert to analog channel. In this example, channels beginning with SPK are considered as spike channels. Channels beginning with LFP and ANC are used as continuous signal channels. Channels beginning with CH are transformed to analog channels. Channels beginning with DIG are used for events conversion, but are not stored in output file. The other channels are ignored.
electrodes SPK1.-1 SPK2.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1

SPK3.-1 SPK4.-1 LFP1.-1 LFP2.-1 LFP3.-1 LFP4.-1 ANC1.-1 ANC2.-1 CH2_T1.-1 CH2_T2.-1 CH2_T3.-1 CH3_T1.-1	if the channel is not defined in elec.dat . The order should be the same as in the input <i>Alpha-Map</i> file.
--	---

- **Example**

Reading information about a file *A2209004.map* :
alphamap2eeg *A2209004.map*

Output to screen :
ALPHAMAP2EEG : V1.04 14-03-2008
File conversion from the Alpha-Map format to ELAN eeg format.

Reading Alpha-Map file
A2209004.map...
Alpha-Map file version : 85
#chan name #Alpha type f(kHz) Gain #samples duration (s)
1 2-AI001 20001 0 12.500 5.000 66501 5.320080
2 2-AI008 20008 0 12.500 5.000 66501 5.320080
3 2-AI017 20017 0 12.500 5.000 66501 5.320080
4 2-AI024 20024 0 12.500 5.000 66501 5.320080
done.
ALPHAMAP2EEG done.

- **Comments**

- If *myeegfile* and *myparfile* are omitted, the *Alpha-Map* file is read. The information about all channels are displayed.
- The output sampling frequency is the same for all channels. It is the highest sampling frequency of the input continuous channels.

- **Current version**

1.06 05-04-2016

- **History**

- 1.00 10-08-2005 (PEA) : 1st version.
- 1.04 14-03-2008 (PEA) : conversion of digital channels to analog channels added (type 3).
- 1.05 06-06-2008 (PEA) : better use data dynamic (physical maximum and minimum).
- 1.06 05-04-2016 (PEA) : new HDF5 ELAN file format.

- **Files**

\$ELANPATH/bin/alphamap2eeg

- **See also**

ascii2eeg

- **Description**

Conversion tool for ASCII (text) files to EEG format. ASCII file must contain multiplexed data where each column correspond to a channel and each line a temporal sample.

- **Usage**

ascii2eeg myasciifile myeegfile myparfile

with :

- myasciifile : ASCII eeg filename (with extension).
- myeegfile : output EEG file (no extension).
- myparfile : text file containing conversion parameters (contains sampling frequency, electrode list).

- **Fields of parameter file and example**

multiplying_coef 0.25	Multiplying coefficient to apply to data.
sampling_frequency 128	Sampling frequency of the signal (Hz).

electrodes Fp1.18 Fp2.19 F7.13 F3.14 Fz.15 F4.16 F8.17 T3.8 C3.9 Cz.10 C4.11 T4.12 T5.3 P3.4 Pz.5 P4.6 T6.7 O1.1 O2.2 A1.-1 A2.-1 Foto.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order should be the same as in the input ASCII file.
---	--

- **Example**

- **Comment**

- Events may be added by converting event files to Elan event file (.pos).

- **Current version**

1.03 23-11-2012

- **History**

- 1.00 30-08-2005 (PEA) : 1st version.
- 1.01 13-08-2007 (PEA) : minor modification.
- 1.02 22-11-2012 (PEA) : fixes an error when writing data (bad byte order PC-style).
- 1.03 23-11-2012 (PEA) : changes multiplying_coef to multiplying_coef field of parameter file.

- **Files**

\$ELANPATH/bin/ascii2eeg
 \$ELANPATH/bin64/ascii2eeg

- **See also**

biosemi2eeg

- **Description**

Conversion tool for Biosemi files to EEG format.

- **Usage**

biosemi2eeg mybiosemifile.bdf myeegfile [myparfile]

with :

- mybiosemifile.bdf : Biosemi file to convert (with extension).
- myeegfile : output EEG file (no extension).
- option :
 myparfile : text file containing conversion parameters (electrode names and numbers). If this file is omitted, names are read from the Biosemi file and the first entry corresponding in elec.dat is chosen.

- **Fields of parameter file and example**

electrodes P1.21 P2.22 H1.1091 H2.1092 A1.225 A2.226 Status.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order should be the same as in the input Biosemi file.
---	--

- **Example**

- **Comments**

- **Current version**

1.05 18-11-2011

- **History**

- 1.00 09-02-2007 (PEA) : 1st version.
- 1.01 09-02-2007 (PEA) : data conversion running. Still problems with triggers.
- 1.02 23-01-2008 (PEA) : conversion of data and trigger running. Same results with [EEGLAB](#) ^[34] ([BioSig](#) ^[35]).
- 1.03 18-11-2010 (PEA) : correction for event channel with baseline different from 0.
- 1.04 04-05-2011 (PEA) : adds more precision (10 decimals) in for saving sampling period in output header file (for power of 2 sampling frequencies).
- 1.05 18-11-2011 (PEA) : fixes event detection (low or high 8 bits).

- **Files**

\$ELANPATH/bin/biosemi2eeg

- **See also**

brainamp2eeg

- **Description**

Conversion tool for Brainamp files to EEG format.

- **Usage**

brainamp2eeg mybrainamp_headerfile.vhdr myeegfile [myparfile]

with :

- mybrainamp_headerfile.vhdr : Brainamp header file (with extension). It contains the names of data and event files (.eeg et .vmrk).
- myeegfile : output EEG file (no extension).
- option :
myparfile : text file containing conversion parameters.

- **Fields of parameter file and example**

electrodes P1.21 P2.22 H1.1091 H2.1092 A1.225 A2.226 Cable.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order should be the same as in the input Brainamp file. This field is optional . If omitted, the name of the channels is read from the Brainamp file. The first entry corresponding to the label in elec.dat is choosen.
nb_event_code 6	Number of different event code to use from the VMRK event file. This field is optional . If omitted, events of type "Stimulus" only are converted. If this field is present, the fields "event_names" and "event_codes" are required. If not, they are unnecessary.
event_names walk slow fast A B C	List of event types used in the VMRK event file. These labels will be associated to the digital codes defined with "event_codes". This field is required when the field "nb_event_code" is present.
event_codes 1 2 3 4 5 6	List of event codes to associate to the previous names. The order must be the same as the name order. The stimulus code of the VMRK file is added to these codes (see Example section). This field is required when the field "nb_event_code" is present.

- **Example**

- Marker file with different kind of stimulus :

The following files contains 3 kinds of triggers : Stim, Video, Response. Each type of event may have different values :

Stim : always 1
Response : 1 or 2

Video : 1 to 15

```
Brain Vision Data Exchange Marker File, Version 1.0

[Common Infos]
Codepage=UTF-8
DataFile=tap0003.eeg

[Marker Infos]
; Each entry: Mk=,,
; ,
; Fields are delimited by commas, some fields might be omitted (empty).
; Commas in type or description text are coded as "\1".
Mk1=New Segment,,1,1,0,20110531174504332097
Mk2=Comment,actiCAP not connected,1,1,0
Mk3=Video,V 5,6193,1,0
Mk4=Stim,S 1,7693,1,0
Mk5=Response,R 1,8118,1,0
Mk6=Video,V 4,9571,1,0
Mk7=Stim,S 1,11070,1,0
Mk8=Response,R 1,11446,1,0
Mk9=Video,V 14,13048,1,0
Mk10=Stim,S 1,14547,1,0
Mk11=Response,R 2,15041,1,0
Mk12=Video,V 1,16814,1,0
Mk13=Stim,S 1,18314,1,0
Mk14=Response,R 1,18642,1,0
Mk15=Video,V 13,20571,1,0
Mk16=Stim,S 1,22071,1,0
Mk17=Video,V 4,23968,1,0
Mk18=Stim,S 1,25468,1,0
Mk19=Response,R 1,25771,1,0
Mk20=Video,V 13,27695,1,0
Mk21=Stim,S 1,29195,1,0
Mk22=Response,R 2,29621,1,0
Mk23=Video,V 5,31192,1,0
Mk24=Stim,S 1,32692,1,0
Mk25=Response,R 1,32987,1,0
Mk26=Video,V 10,34749,1,0
Mk27=Stim,S 1,36248,1,0
Mk28=Response,R 2,36516,1,0
Mk29=Video,V 2,38286,1,0
Mk30=Stim,S 1,39785,1,0
Mk31=Response,R 1,40161,1,0
Mk32=Video,V 14,41723,1,0
```

To import all these events, we use the following parameter file (this is the event import part only):

```
nb_event_code 3
event_names Video Stim Response
event_codes 100 0 200
```

100 is added to each Video code, 0 to each Stim code and 200 to each Response code.
The resulting event file (after use of [eegpos](#) ^[36]) is :

```
1 255 0
6193 105 0
7693 1 0
8118 201 0
9571 104 0
11070 1 0
11446 201 0
13048 114 0
14547 1 0
15041 202 0
16814 101 0
18314 1 0
18642 201 0
20571 113 0
22071 1 0
23968 104 0
25468 1 0
25771 201 0
27695 113 0
29195 1 0
29621 202 0
31192 105 0
32692 1 0
32987 201 0
34749 110 0
36248 1 0
36516 202 0
38286 102 0
39785 1 0
40161 201 0
```

41723 114 0

The first event (255) corresponds to the New Segment event of Brainamp. The Comment event is removed.

- **Comments**

- If the parameter file is omitted, names are read from the Brainamp file and the first entry corresponding in elec.dat is chosen. If there is no corresponding name in elec.dat, the number is -1.
- The Brainamp header file contains the names of the data and event files (.eeg and .vmrk). The output Elan EEG file contains converted data and events.
- Warning : the Brainamp data file and the Elan data file have the same extension (.eeg). Be careful to use different names when converting Brainamp file to Elan.
- Note to INSERM U821 users : ActiCap is based on 10/10 system. When converting Brainamp files, use the following electrode positions (listed in the file [elecActiCap.dat](#) ⁽³⁷⁾) :

electrodes
 Fp1.2176
 Fp2.2177
 F7.2178
 F3.2179
 Fz.2180
 F4.2181
 F8.2182
 FC5.2183
 FC1.2184
 FC2.2185
 FC6.2186
 T7.2187
 C3.2188
 Cz.2189
 C4.2190
 T8.2191
 TP9.2221
 CP5.2192
 CP1.2193
 CP2.2194
 CP6.2195
 TP10.2227
 P7.2197
 P3.2198
 Pz.2199
 P4.2200
 P8.2201
 PO9.2203
 O1.2204
 Oz.2237
 O2.2205
 PO10.2206
 EOG.-1
 EMGL.-1
 EMGR.-1

- **Current version**

1.10 08-06-2011

- **History**

- 1.03 19-07-2006 (PEA) : Brainamp file format evolution.
- 1.04 19-09-2006 (PEA) : adding fields in parameter files to use events that are not of type "Stimulus" (type defined by Brainamp).
- 1.05 21-09-2006 (PEA) : repair error with event codes for events of type different from "Stimulus".
- 1.06 22-09-2006 (PEA) : sort events by time before conversion (comments are not sorted by time).
- 1.08 05-03-2008 (PEA) : repair error with path of header file (vhdr). This file contains the names without path of the data and event files.
- 1.09 20-12-2010 (PEA) : fixes memory allocation (initialization of array of samples for events and free of index of channels in elec.dat).
- 1.10 08-06-2011 (PEA) : changes stimulation codes according to Brainamp.

- **Files**

\$ELANPATH/bin/brainamp2eeg

- **See also**

Attachment	Size
elecActiCap.dat ⁽³⁸⁾	305 bytes

coil2mat

- **Description**

Conversion function for Elan coil files (.coil) to Matlab.

- **Usage**

[coil_diameter, coil_number, channel_number, coil_coord, coil_orient, coil_weight] = coil2mat(coil_filename, varargin)
with :

- coil_file_name : input filename (with extension) with the complete path.
- option :
 'save', mat_name : to save all the results in a .mat file named mat_name.mat .

output :

- coil_diameter : diameter of the coils (mm).
- coil_number : number of coils.
- channel_number : number of channels.
- coil_coord : coil coordinates.
- coil_orient : coil orientations.
- coil_weight : coil weights.

- **Fields of parameter file and example**

- **Example**

- **Comments**

- **Current version**

1.00 05-03-2010

- **History**

- 1.00 05-03-2010 (PEA) : first version.

- **Files**

\$ELANPATH/misc/matlab/coil2mat.m

- **See also**

ctf275_2coil

- **Description**

Conversion tool for coordinates of sensors (VSM-CTF Omega 275 channels) to Elan .coil format (text file).

- **Usage**

ctf275_2coil myctfdirectory.ds mycoilfile [-withREF]

with :

- myctfdirectory.ds : CTF dataset directory (with extension).
- mycoilfile : output coordinates file (with extension .coil).
- option :
 -withREF : if present, convert coordinates of REF-SENS and MEG-SENS channels. If omitted, only convert MEG-SENS channels.

- **Coil file format and examples**

- Coil file version 1

It is a text file with a .coil suffix. The informations are identified by keywords and values :

diametre_coils : coil diameters (mm)

nb_coils : number of coils

nb_voies : number of channels

: comment(example : #channel 1 : A1)

voieN : number of coils for channel N with N is the indice of the channel (starting from 1). For example : *voie1 2* means 2 coils for channel 1.

C_N_M : coordinates, orientation and weight of coil M of channel N. C_N_M is followed by Xcoil Ycoil Zcoil MXcoil MYcoil MZcoil WCoil with :

Xcoil, Ycoil, Zcoil : coordinates of coil (mm) in head axis system.

MXcoil, MYcoil, MZcoil : orientation of coil (mm). These values must be normalized.

Wcoil : weight of coil.

Download examples : [exemple_bti.coil](#) [39], [exemple_ctf.coil](#) [40], [exemple_Neuromag.coil](#) [41].

Exemple_ctf.coil :

```
diametre_coils 20.000000
nb_coils 286
nb_voies 143
#voieN nb_coils
#C_N_M Xcoil Ycoil Zcoil MXcoil MYcoil MZcoil PoidsCoil
#N = channel indice, M = coil indice

#voie 1 LC11
voie1 2
C_1_1 78.603996 81.996788 106.587189 -0.788269 -0.442448 -0.427636 1
C_1_2 118.027443 104.124817 127.974411 0.788269 0.442448 0.427636 1

#voie 2 LC12
voie2 2
C_2_1 85.594002 56.471432 114.443710 -0.858740 -0.263944 -0.439203 1
C_2_2 128.541901 69.671989 136.409454 0.858740 0.263944 0.439203 1

.....

#voie 143 RP45
voie143 2
C_143_1 -17.488001 -79.049843 115.331032 0.154553 0.854542 -0.495853 1
C_143_2 -25.217600 -121.787811 140.129990 -0.154553 -0.854542 0.495853 1
```

Last modifications :

- squid changed to coil : 05-07-1997 (Anne Cheylus, Olivier)
- 1st version : 21-01-1997 (Anne, Olivier)

- Coil file version 2

It is also a text file with a .coil suffix. The name of file must contain V2 to make difference with version 1. It is faster to read.

As for version 1, the informations are identified by keywords and values :

```
diametre_coils : coil diameters (mm)
nb_coils : number of coils
nb_voies : number of channels
# : comment(example : #channel 1 : A1)
voieN : number of coils for channel N (with N is the indice of the channel (starting from 1)), weight of each coil,
indice of each coil. For example : voie1 2 1 -1 3 4 means 2 coils for channel 1 (coils 3 and 4), the weight of coils 3
and 4 are respectively 1 and -1. Channel 1 is the result of the difference of coil 3 minus coil 4.
coil_data : matrix of coordinates and orientations of coils. Xcoil, Ycoil, Zcoil must be in mm. MXcoil, MYcoil, MZcoil
must be normalize.
Xcoil1 Ycoil1 Zcoil1 MXcoil1 MYcoil1 MZcoil1
Xcoil2 Ycoil2 Zcoil2 MXcoil2 MYcoil2 MZcoil2
XcoilN YcoilN ZcoilN MXcoilN MYcoilN MZcoilN
Download example : ctf151\_V2.coil [42].
```

Last modifications :

coils version 2 : march 2000 (Anne Cheylus)

- **Examples**

- **Comments**

ctf275_2coil creates version 1 coil file.

- **Current version**

1.02 24-06-2008

- **History**

- 1.01 (03-06-2008) : Prise en compte de la polarité des coils => Poids = -1 si ProperGain > 0
- 1.02 (24-06-2008) : Suppression bug lecture du nom du fichier *.ds
[modifier]
 - 1.01 03-06-2008 (MM) : use polarity of coils (weight=-1 if ProperGain > 0).
 - 1.02 24-06-2008 (MM) : bug fixe when reading name of ds directory.

- **File**

\$ELANPATH/bin/ctf275_2coil

- **See also**

Attachment	Size
exemple_bti.coil ^[39]	5.95 KB
exemple_ctf.coil ^[40]	22.99 KB
exemple_Neuromag.coil ^[41]	23.38 KB

- **Description**

Conversion tool for CTF MEG files to EEG format.

- **Usage**

ctf2eeg myctfdirectory.ds myparfile myeegfile [-v]

with :

- myctfdirectory.ds : CTF dataset directory (with extension).
- myparfile : text file containing conversion parameters (electrode names and numbers in elec.dat).
- myeegfile : output EEG file (no extension).
- option :
 - v : verbose mode. If omitted, verbose mode is off.

- **Fields of parameter file and example**

electrodes	Output channel name list :
UPPT.-1	
SCLK.-1	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
BG1.-1	
BG2.-1	
BG3.-1	The order and number of channels should be the same as in the input CTF res4 file.
etc...	

- **Example**

- **Comments**

- Note to CTF 275 MEG in Lyon users : a parameter file is available to download : [ctf275.par](#) ^[43]
- Events are read from channels with label beginning with "STIM", "UPPT", "USPT", "UDIO". The file MarkerFile is not used. It can be converted with ctfmrk2pos to Elan event file.

- **Current version**

1.15 04-05-2011

- **History**

- 1.04 12-07-2005 (PEA) : new CTF file definition (MEG42).
- 1.05 11-10-2005 (PEA) : corrections for file size > 2 GB.
- 1.06 16-01-2006 (PEA) : corrections for file size > 2 GB.
- 1.07 18-01-2006 (PEA) : use of sensorType (read in res4) to define units of the sensor (MEG, EEG...).
- 1.08 18-01-2006 (PEA) : test for different gradient orders for MEG channels.
- 1.09 22-02-2006 (PEA) : repair error for events in "old" files (LENA).
- 1.10 27-04-2006 (PEA) : repair error when reading multiple data files (> 2 GB). Forgot to read 8 first bytes.
- 1.11 13-08-2007 (PEA) : minor modification.
- 1.12 10-09-2009 (PEA) : bug fix when a number of stimulation channels is used (symptom : repeated event code).
- 1.13 10-09-2009 (PEA) : correction of type definition in MegDefs.h for 64 bits compilation.
- 1.14 23-11-2010 (PEA) : correction in rchmpl (addition of maximum length line) in libgen.
- 1.15 04-05-2011 (PEA) : adds more precision (10 decimals) in for saving sampling period in output header file.

- **Files**

\$ELANPATH/bin/ctf2eeg

- **See also**

ctf2ep, ctfmrk2pos, pos2ctfmrk

Attachment Size

[ctf275.par](#) ^[44] 5.67 KB

ctf2ep

- **Description**

Conversion tool for CTF MEG files to Elan evoked potential format (.p).

- **Usage**

ctf2eeg myctfdirectory.ds myparfile myepfile [+v]

with :

- myctfdirectory.ds : CTF dataset directory (with extension).

- myparfile : text file containing conversion parameters (electrode names and numbers in elec.dat, flag for channel conversion).
- myepfile : output evoked potential file (no extension).
- option :
 - +v : verbose mode. If omitted, verbose mode is off.

- **Fields of parameter file and example**

channels	Output channel name list :
STIM.-1 0	
BG1.-1 0	The format is "name.number conversion_code", where :
BG2.-1 0	name is the name of the channel in elec.dat
BG3.-1 0	number is the rank of the channel in elec.dat
MLC11.785 1	conversion_code indicates whether to convert the channel or not. Possible values are :
MLC12.786 1	0 channel is ignored
MLC13.787 1	1 is stored
MLC14.788 1	2 is stored and data is set to 0
MLC15.789 1	
MLC21.790 1	The order and number of channels should be the same as in the input CTF res4 file.
MLC22.791 1	
MLC23.792 1	

- **Examples**

- Example 1 – To create an *.p file with MEG channels only

Example of parameter file with channel specifications :

```
channels
STIM.-1 0
BG1.-1 0
BG2.-1 0
BG3.-1 0
BP1.-1 0
BP2.-1 0
P22.-1 0
```

...

```
R13.-1 0
R22.-1 0
R23.-1 0
MLC11.785 1
MLC12.786 1
MLC13.787 1
MLC14.788 1
MLC15.789 1
MLC21.790 1
MLC22.791 1
```

...

```
MZO01.932 1
MZO02.933 1
MZO01.934 1
MZO02.935 1
FP1.18 0
FPZ.39 0
FP2.19 0
F7.13 0
F3.14 0
FZ.15 0
F4.16 0
```

...

```
P5.20 0
P6.23 0
CZ.10 0
```

- Example 2 – To create an *.p file with EEG channels only
- Example of parameter file with channel specifications

```
channels
STIM.-1 0
BG1.-1 0
BG2.-1 0
BG3.-1 0
BP1.-1 0
BP2.-1 0
P22.-1 0
```

...

```
R13.-1 0
R22.-1 0
R23.-1 0
MLC11.785 0
MLC12.786 0
MLC13.787 0
MLC14.788 0
```

```
MLC15.789 0
MLC21.790 0
MLC22.791 0
```

...

```
MZO01.932 0
MZO02.933 0
MZP01.934 0
MZP02.935 0
FP1.18 1
FPZ.39 1
FP2.19 1
F7.13 1
F3.14 1
FZ.15 1
F4.16 1
```

...

```
P5.20 1
P6.23 1
CZ.10 1
```

- **Comments**

- **Current version**

1.08 23-11-2009

- **History**

- 1.04 10-08-2005 (PEA) : new CTF file definition (MEG42).
- 1.05 11-10-2005 (PEA) : corrections for file size > 2 GB.
- 1.07 10-09-2009 (PEA) : correction of type definition in MegDefs.h for 64 bits compilation.
- 1.08 23-11-2010 (PEA) : correction in rchmpl (addition of maximum length line) in libgen.

- **Files**

```
$ELANPATH/bin/ctf2ep
$ELANPATH/bin64/ctf2ep
```

- **See also**

ctf2eeg, ctfmrk2pos, pos2ctfmrk

ctfmrk2pos

- **Description**

Conversion tool for CTF marker files to Elan event format (.pos).

- **Usage**

```
ctfmrk2pos myctfdirectory.ds file.out.pos file.conv.par file.out.conv.par
```

with :

- myctfdirectory.ds : CTF dataset directory (with extension) containing the marker file.
- file.out.pos : name of the output file containing the events.
- file.conv.par : name of the input file containing the parameters for converting marker file to pos file.
- file.out.conv.par : name of the output file containing the parameters for converting back pos file to marker file with pos2ctfmrk.

- **Fields of parameter file and example**

conv_nb_code 4	Number of CTF event classes to convert.
conv_name_list start stdsvdev std dev	List of event class names for CTF. WARNING : the names must be strictly the same in MarkerFile.mrk .
conv_pos_code_list 10 20 30 40	List of codes for each CTF event class. These codes are saved in POS file.

- **Example**

- **Comments**

- **Current version**

1.09 23-11-2010

- **History**

- 1.00 10-01-2006 (PEA) : first version.
- 1.01 10-01-2006 (PEA) : minor modification.
- 1.02 14-03-2006 (PEA) : addition of the parameter "conv_name_list" to convert classes to codes. The indice of classes is not relevant for CTF because it may change.
- 1.03 16-03-2006 (PEA) : minor modification.
- 1.04 26-09-2006 (PEA) : blank suppression in parameter file.
- 1.06 07-04-2008 (PEA) : check for memory leaks.
- 1.07 07-10-2008 (PEA) : strict check of names of event classes. Adding epoched acquisition mode support with prestimulus samples.
- 1.08 10-09-2009 (PEA) : correction of type definition in MegDefs.h for 64 bits compilation.
- 1.09 23-11-2010 (PEA) : correction in rchmpl (addition of maximum length line) in libgen.

- **Files**

\$ELANPATH/bin/ctfmrk2pos
\$ELANPATH/bin64/ctfmrk2pos

- **See also**

ctf2eeg, ctf2ep, pos2ctfmrk

ctfssv2ep

- **Description**

Conversion tool for CTF SSV files to Elan evoked potential format (.p).

- **Usage**

ctfssv2ep myssvfile.ssv myparfile myepfile [+xml_file xml_filename] [+v]

with :

- myctfdirectory.ds : CTF dataset directory (with extension).
- myparfile : text file containing conversion parameters (electrode names and numbers in elec.dat).
- myepfile : output .p file (no extension).
- options :
 - +xml_file xml_filename : XML filename to save matrix in (with extension).
 - +v : verbose mode. If omitted, verbose mode is off.

myparfile lists the name and number of all the sensors of the dataset.

- **Fields of parameter file and example**

electrodes UPPT.-1 SCLK.-1 BG1.-1 BG2.-1 BG3.-1 etc...	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order and number of channels should be the same as in the input CTF res4 file.
nb_tot_chan 340	Number of channels in CTF dataset.

- **Example**

- **Comments**

- All channels of the dataset must be listed in the parameter file, because SSV file uses the indice of channels.

- **Current version**

1.04 23-11-2010

- **History**

- 1.00 18-02-2010 (PEA) : first version.
- 1.01 18-02-2010 (PEA) : correction for channels/vectors order.
- 1.02 22-02-2010 (PEA) : adding option +xml_file to save a XML text file for using with eegproject, eegfiltica and matrix2p.

- 1.03 24-02-2010 (PEA) : adding the field `nb_tot_chan` in parameter file. It gives the number of channel of an Elan EEG file that may be used with `eegproject` and `eegfiltica`. The value is the number of analog channels (without the 2 digital channels of Elan files `dig1` and `dig2`).
- 1.04 23-11-2010 (PEA) : correction in `rchmpl` (addition of maximum length line) in `libgen`.

- **Files**

```
$ELANPATH/bin/ctfssv2ep
$ELANPATH/bin64/ctfssv2ep
```

- **See also**

`ctf2eeg`, `ctf2ep`, `ctfmrk2pos`, `pos2ctfmrk`, `eegproject`, `eegfiltica`, `matrix2p`

daq2eeg

- **Description**

Conversion function for Matlab DAQ file (acquisition toolbox) to ELAN EEG format.

- **Usage**

```
daq2eeg(daq_filename, eeg_filename, channel_list)
```

with :

- `daq_filename` : input **.daq** filename with the complete path.
- `eeg_filename` : output **.eeg** filename with the complete path.
- `channel_list` : channel names and indices in `elec.dat` list (for example 'Cz.10' or 'chan1.-1' for channel not defined in `elec.dat`).

- **Fields of parameter file and example**

- **Example**

Matlab code for file `test.daq` containing 5 channels (not listed in `elec.dat` and named `chan1` to `chan5`) :

```
for i=1:5
chan_list{i}=['chan' num2str(i) '-1']; % Channel list generation
end
daq_filename = 'test.daq';
eeg_filename = 'test_elan.eeg';
daq2eeg(daq_filename, eeg_filename, channel_list);
```

- **Comments**

- **Current version**

1.00 20-04-2011

- **History**

- 1.00 20-04-2011 (PEA) : 1st version.

- **Files**

```
$ELANPATH/misc/matlab/daq2eeg.m
```

- **See also**

[mat2eeg](#) ^[2], [eeg2mat](#) ^[1]

edf2eeg

- **Description**

Conversion tool for EDF ([European Data Format](#) ^[45]) files to EEG format.

- **Usage**

```
edf2eeg myEDF_file.edf myparfile myeegfile
```

with :

- myEDF_file.edf : EDF file to convert (with extension).
- myparfile : text file containing conversion parameters (electrode names and numbers in elec.dat, rejected channels).
- myeegfile : output EEG file (no extension).

- **Fields of parameter file and example**

electrodes P1.21 P2.22 H1.1091 H2.1092 A1.225 A2.226 Status.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order and number of channels should be the same as in the input EDF file.
reject_channel_nb 2	Channel list to reject (not converted). If omitted, all channel are converted.
reject_channel_list 49 50	Number of channel to reject (not converted). This field is required if the field "reject_channel_nb" is present.

- **Example**

An EDF file (trial.edf) has 4 channels. Channels 1-2 use a sampling frequency of 1000Hz, and channels 3-4 use 2000Hz.

The first channels will be converted with the following parameter file (edf1000Hz.par) :

```
electrodes
chan1.-1
chan2.-1
chan3.-1
chan4.-1
reject_channel_nb 2
reject_channel_list 3 4
```

And the command line :

```
edf2eeg trial.edf edf1000Hz.par trial.1000Hz
```

The last channels will be converted with the following parameter file (edf2000Hz.par) :

```
electrodes
chan1.-1
chan2.-1
chan3.-1
chan4.-1
reject_channel_nb 2
reject_channel_list 1 2
```

And the command line :

```
edf2eeg trial.edf edf2000Hz.par trial.2000Hz
```

These 2 command lines will create the output files trial.1000Hz.eeg (and trial.1000Hz.eeg.ent) and trial.2000Hz.eeg (and trial.2000Hz.eeg.ent). Each one have only 2 channels (one with channels 1 and 2, and the other with channels 3 and 4).

- **Comments**

- Elan supports only one sampling frequency in an EEG file as EDF files may have different sampling frequencies per channel in one file. In this case, one may convert only channels with the same sampling frequency (see fields reject_channel_nb and reject_channel_list of the parameter file), and create as many Elan files as different sampling frequencies in the EDF file. See example above.
- Elan converts only data. No event are stored in EDF file. Events may be added by converting event files to Elan event file (.pos).

- **Current version**

1.04 25-09-2015

- **History**

- 1.00 16-02-2010 (PEA) : first version.
- 1.01 25-03-2010 (PEA) : fixes output data allocation.
- 1.02 29-05-2012 (PEA) : fixes an error at beginning of data.
- 1.03 20-07-2015 (PEA) : read and convert the annotation triggers (only annotations with numbers, no string).
- 1.04 25-09-2015 (PEA) : fixes crash when no annotation triggers.

- **Files**

\$ELANPATH/bin/edf2eeg

- **See also**

- **Description**

Conversion function for Elan EEG files (.eeg and .eeg.ent) to [Fieldtrip](#) ^[46].

- **Usage**

[ft_hdr, ft_events, ft_data, ft_sensors] = eeg2fieldtrip(InEEGFile, varargin)
with :

- InEEGFile : input filename (with extension) with the complete path.
- 'eventfile', eventfilename: Elan event file (.pos) to convert. If omitted, no event is imported.
- 'sensorfile', sensorfilename: sensor coordinates file. For CTF, the file is .res4 of the dataset. See Fieldtrip help of ft_read_sens for more informations.
- 'sensorfilebox': opens a dialog box to select sensor file. If omitted and 'sensorfile' is omitted, no sensor coordinates is imported.
- 'chanindx', chanindx : selected channel indices. If omitted, all channels are selected.
- 'begsample', begsample : first sample of data. If omitted, begsample = 1 (1st sample of file).
- 'endsample', endsample : last sample of data (or 'all' for last sample of file). If omitted, endsample = last sample of file.
- 'save', filename : saves ft_hdr, ft_events, ft_data, ft_sensors to filename MAT file. If omitted, these variables are not saved.

output :

- ft_hdr : Fieldtrip header.
- ft_events : events stored in Fieldtrip event structure.
- ft_data : data stored in Fieldtrip data structure.
- ft_sensors : sensor (EEG or MEG) positions in Fieldtrip data structure.

- **Fields of parameter file and example**

- **Example**

- **Comments**

- **Current version**

1.00 10-02-2012

- **History**

- 1.00 10-02-2012 (PEA) : 1st version.

- **Files**

\$ELANPATH/misc/matlab/eeg2fieldtrip.m
\$ELANPATH/misc/matlab/eeg2mat.m

- **See also**

[eeg2mat](#) ^[1], [mat2eeg](#) ^[2], [ep2fieldtrip](#) ^[24]

eeg2mat

- **Description**

Conversion function for Elan EEG files (.eeg and .eeg.ent) to Matlab.

- **Usage**

[m_data, m_events, v_label_selected, s_fs, s_nb_samples_all, s_nb_channel_all, v_label_all, v_channel_type_all, v_channel_unit_all, str_ori_file1, str_ori_file2] = eeg2mat(filename, s_sample_start, s_sample_stop, v_channel_list, varargin)
with :

- filename : input filename (with extension) with the complete path.
- s_sample_start : sample number at the beginning of reading window (1 to start at the first sample).
- s_sample_stop : sample number at the end of the reading window ('all' to select all samples after s_sample_start).
- v_channel_list : vector of selected channels (rank in the eeg file) ('all' to select all channels).
- option :
 'save', mat_name : to save all the results in a .mat file named mat_name.mat .

output :

- m_data : data recorded by the selected channels (units as in the .eeg file), 1 channel per line, 1 sample per column.
- m_events : table with the events (samples and event codes).
- v_label_selected : name of the selected channels.

- `s_fs` : sampling frequency (Hz).
- `s_nb_samples_all` : total number of samples per channel in eeg file.
- `s_nb_channel_all` : total number of channels.
- `v_label_all` : name of all channels (with numbers in elec.dat).
- `v_channel_type_all` : sensor types (EEG, MEG, ...).
- `v_channel_unit_all` : sensor units (μ V, fT, ...).
- `str_ori_file1` : acquisition system.
- `str_ori_file2` : acquisition place.

- **Fields of parameter file and example**

- **Example**

See example of [mat2eeg](#) [2].

- **Comments**

- **Current version**

1.05 09-08-2010

- **History**

- 1.05 09-08-2010 (PEA) : minor change.

- **Files**

\$ELANPATH/misc/matlab/eeg2mat.m

- **See also**

[mat2eeg](#) [2]

elan2eeglab

- **Description**

Matlab conversion function for Elan continuous (**.eeg** and **.eeg.ent**) or evoked potential (**.p**) data file to [EEGLAB](#) [34] file format (**.set**).

- **Usage**

```
[EEG] = elan2eeglab(ElanFile, EeglabFile)
or
[EEG] = elan2eeglab(ElanFile, EeglabFile, EventFile)
with :
```

- ElanFile : input filename (with extension) with the complete path (may be in continuous data or evoked potential format).
- EeglabFile :
- EventFile : Elan event file (.pos) to convert.
If omitted :
- for evoked potential file format (**.p**), the event used is stimulus.
- for continuous file format (**.eeg**), the event are read from file.

output :

- EEG : EEGLAB structure.

- **Fields of parameter file and example**

- **Example**

- **Comments**

- **Current version**

1.00 06-09-2012

- **History**

- 1.00 06-09-2012 (PEA) : 1st version.

- **Files**

\$ELANPATH/misc/matlab/elan2eeglab.m
\$ELANPATH/misc/matlab/eeg2mat.m
\$ELANPATH/misc/matlab/ep2mat.m

- **See also**

[eeg2mat](#) ^[1], [mat2eeg](#) ^[2], [ep2mat](#) ^[3]

ep2fieldtrip

- **Description**

Conversion function for Elan EP files (.p) to [Fieldtrip](#) ^[46].

- **Usage**

ft_data = ep2fieldtrip(OutFileFT, varargin)
with :

- OutFileFT : output Fieldtrip filename (with path, no extension .mat).
- option varargin :
PFileList : Elan .p file(s).

output :

- ft_data : Fieldtrip structure.

- **Fields of parameter file and example**

- **Examples**

- **Comments**

- It creates a simple Fieldtrip structure with labels, sampling frequency, and data. Channel coordinates need to be imported.

- **Current version**

1.02 10-08-2010

- **History**

- 1.00 22-07-2010 (PEA) : first version.
- 1.01 10-08-2010 (PEA) : change in variable names.
- 1.02 10-08-2010 (PEA) : accept PFileList as cell array.

- **Files**

\$ELANPATH/misc/matlab/ep2fieldtrip.m
\$ELANPATH/misc/matlab/ep2mat.m

- **See also**

[ep2mat](#) ^[3], [mat2ep](#) ^[4], [eeg2fieldtrip](#) ^[47]

ep2mat

- **Description**

Conversion function for Elan EP files (.p) to Matlab.

- **Usage**

[HEADER1, HEADER2, DATA, CHANNELS] = ep2mat(p_file_name, varargin)
with :

- p_file_name : input filename (with extension) with the complete path.
- option :
'save', mat_name : to save all the results in a .mat file named mat_name.mat .

output :

- HEADER1 : structure containing the first header of the ep file.

- HEADER1.s_Version : file version (-1 or -3).
- HEADER1.s_Header_Size : header size in bytes.
- HEADER1.s_Event_Code : averaged event code.
- HEADER1.v_reserved : reserved.
- HEADER2 : structure containing second header of the ep file.
 - HEADER2.s_Nb_Channels : total number of channels.
 - HEADER2.s_Nb_Sample_per_Channel : number of samples per channel.
 - HEADER2.s_Time_Epoch : duration (ms).
 - HEADER2.s_Nb_Sample_PreStim : number of samples for prestimulus.
 - HEADER2.s_Sampling_Period : sampling period (ms).
 - HEADER2.s_Min_Sig_Value : minimum signal value (amplitude unit).
 - HEADER2.s_Max_Sig_Value : maximum signal value (amplitude unit).
 - HEADER2.v_Elec : channel numbers in elec.dat.
 - HEADER2.v_Triplets : channel spherical coordinates (r, theta, phi).
 - HEADER2.s_Nb_Event_Aver : number of averaged epochs.
 - HEADER2.s_Nb_Samp_Inhib_Artef_Rej : number of inhibition samples for artefact rejection.
 - HEADER2.s_Flag_Artef_Rej : flag for the artefact rejection.
 - HEADER2.s_Flag_Baseline_Correction : flag for the baseline correction (=0 no correction, other = event code used for the correction).
 - HEADER2.s_Amplifier_Gain : amplifiers gain.
 - HEADER2.s_Low_Cutfrequency : low cut frequency for the amplifiers (Hz).
 - HEADER2.s_High_Cutfrequency : high cut frequency for the amplifiers (Hz).
 - HEADER2.v_Baseline_Value_per_Chan : baseline values for each channel.
 - HEADER2.reserved : reserved.
- DATA : data matrix (HEADER2.s_Nb_Sample_per_Channel x HEADER2.s_Nb_Channels).
- CHANNELS : structure containing channel informations.
 - v_Label : channel labels.
 - v_Theta : channel spherical coordinates theta (degree).
 - v_Phi : channel spherical coordinates phi (degree).
 - v_Type : channel type (EEG, MEG, LFP, unknown,...).
 - v_Units : channel unit (uV, mV, fT, unknown,...).

• **Fields of parameter file and example**

• **Example**

- Read file /data/suj001.p
[HEADER1, HEADER2, DATA, CHANNELS] = ep2mat('/data/suj001.p');
- Read file /data/suj001.p and saves in /data/matfiles/suj001.mat
[HEADER1, HEADER2, DATA, CHANNELS] = ep2mat('/data/Elan/suj001.p', 'save', '/data/matfiles/suj001.mat');

• **Comments**

• **Current version**

1.03 09-08-2010

• **History**

- 1.03 09-08-2010 (PEA) : minor change (variable names).

• **Files**

\$ELANPATH/misc/matlab/ep2mat.m

• **See also**

[mat2ep](#) ^[4]

ep2nutmeg

• **Description**

Conversion function for Elan EP files (.p) to [NUTMEG](#) ^[48].

• **Usage**

nm_data = ep2nutmeg(OutFileNM, data_type, varargin)
with :

- OutFileNM : output NUTMEG filename (with path, no extension .mat).
- data_type: string describing data type. May be 'eeg' or 'meg'. It defines the way to find sensor coordinates (and orientations for MEG).
- option varargin :
PFileName : Elan .p file name.

output :

- nm_struct : NUTMEG structure.

- **Fields of parameter file and example**

- **Examples**

- **Comments**

- Uses SPM for EEG channel coordinates.
- Uses Nutmeg for CTF MEG sensor positions and orientations.

- **Current version**

1.01 11-08-2010

- **History**

- 1.00 10-08-2010 (Sarang Dalal/PEA) : first version.
- 1.01 10-08-2010 (Sarang Dalal/PEA) : import of channel coordinates fro EEG files (with SPM ext1020.sfp template). Import of sensor positions and orientations for CTF MEG data (with Nutmeg).

- **Files**

\$ELANPATH/misc/matlab/ep2nutmeg.m
\$ELANPATH/misc/matlab/ep2mat.m

- **See also**

epeg2spm

- **Description**

Conversion function for Elan EP files (.p) obtained from EEG data to [SPM](#) ^[49].

- **Usage**

D = epeg2spm(OutFileSPM, PathSPM, varargin)
with :

- OutFileSPM : output SPM filename (with path, no extension .mat).
- PathSPM : path to output SPM files.
- option varargin :

PFileList : Elan .p file(s).

'keep_reference' : don't change reference. If omitted, average reference is processed (see output files).

output :

- D : SPM structure.

output files:

- OutFileSPM.mat, OutFileSPM.dat : SPM files without montage informations.
- 'M'OutFileSPM.mat, 'M'OutFileSPM.dat : SPM files with montage informations (with average reference or not).

- **Fields of parameter file and example**

- **Examples**

- Read /data/Elan/suj001.cond1.p and /data/Elan/suj001.cond2.p files and convert to a single SPM structure with 2 conditions
D =epeg2spm('suj001', '/data/SPM/work', '/data/Elan/suj001.cond1.p', '/data/Elan/suj001.cond2.p');
- Read /data/Elan/suj001.cond1.p and /data/Elan/suj001.cond2.p files and convert to a single SPM structure with 2 conditions, keeping reference (no average reference processed) :
D =epeg2spm('suj001', '/data/SPM/work', 'keep_reference', '/data/Elan/suj001.cond1.p', '/data/Elan/suj001.cond2.p');
or
D =epeg2spm('suj001', '/data/SPM/work', '/data/Elan/suj001.cond1.p', '/data/Elan/suj001.cond2.p', 'keep_reference');
- Open a window for Elan .p file(s) selection
D =epeg2spm('suj001', '/data/SPM/work');
- Read /data/Elan/suj001.cond1.p and /data/Elan/suj001.cond2.p files and convert to a single SPM structure with 2 conditions (use a vector of names)
filelist{1}='/data/Elan/suj001.cond1.p';
filelist{2}='/data/Elan/suj001.cond2.p';
D =epeg2spm('suj001', '/data/SPM/work', file_list);

- **Comments**

- This function uses SPM. You need to change the path according to your SPM installation.
- Channel coordinates are read from SPM EEG template ext1020.sfp according to the label.

- **Current version**

1.08 19-03-2012

- **History**

- 1.00 30-04-2010 (PEA) : first version.
- 1.01 03-05-2010 (PEA) : remove CTF dependance.
- 1.02 04-05-2010 (PEA) : adding 'unknown' channels for no label.
- 1.03 22-07-2010 (PEA) : bug fixe for multiple input files.
- 1.04 10-08-2010 (PEA) : change in variable names.
- 1.05 10-08-2010 (PEA) : accept PFileList as cell array.
- 1.06 06-07-2011 (PEA) : fixes sampling frequency test between files.
- 1.07 07-07-2011 (PEA) : fixes duration test between files.
- 1.08 19-03-2012 (PEA) : adds SPM_path variable. Adds average reference option, 'keep_reference' for not changing reference. If omitted, common average reference is computed /li>

- **Files**

\$ELANPATH/misc/matlab/epeg2spm.m
\$ELANPATH/misc/matlab/ep2mat.m

- **See also**

[epmeg2spm](#) ^[27], [ep2mat](#) ^[3], [mat2ep](#) ^[4]

epmeg2spm

- **Description**

Conversion function for Elan EP files (.p) obtained from CTF MEG data to [SPM](#) ^[49].

- **Usage**

D = epmeg2spm(OutFileSPM, PathSPM, CTFres4, varargin)

with :

- OutFileSPM : output SPM filename (with path, no extension .mat).
- PathSPM : path to output SPM files.
- CTFres4 : CTF res4 resource file (with path) corresponding to Elan files.
- option varargin :
PFileList : Elan .p file(s) (from CTF MEG).

output :

- D : SPM structure.

- **Fields of parameter file and example**

- **Examples**

- Read /data/Elan/suj001.cond1.p and /data/Elan/suj001.cond2.p files and convert to a single SPM structure with 2 conditions

```
D =epmeg2spm('suj001', '/data/SPM/work/', '/data/CTF/suj001.ds/suj001.res4', '/data/Elan/suj001.cond1.p',
'/data/Elan/suj001.cond2.p');
```

- Open a window for Elan .p file(s) selection

```
D =epmeg2spm('suj001', '/data/SPM/work/', '/data/CTF/suj001.ds/suj001.res4');
```

- Read /data/Elan/suj001.cond1.p and /data/Elan/suj001.cond2.p files and convert to a single SPM structure with 2 conditions (use a vector of names)

```
filelist{1}='/data/Elan/suj001.cond1.p';
```

```
filelist{2}='/data/Elan/suj001.cond2.p';
```

```
D =epmeg2spm('suj001', '/data/SPM/work/', '/data/CTF/suj001.ds/suj001.res4', file_list);
```

- **Comments**

- This function uses SPM. You need to change the path according to your SPM installation (see lines 37 to 41 of epmeg2spm.m).
- Fiducial coordinates are read in CTF res4 file.

- **Current version**

1.04 10-08-2010

- **History**
 - 1.00 26-04-2010 (PEA) : first version.
 - 1.01 22-07-2010 (PEA) : correction for multiple input files.
 - 1.02 22-07-2010 (PEA) : bug fixe for multiple input files.
 - 1.03 10-08-2010 (PEA) : change in variable names.
 - 1.04 10-08-2010 (PEA) : accept PFileList as cell array.

- **Files**

\$ELANPATH/misc/matlab/epmeg2spm.m
\$ELANPATH/misc/matlab/ep2mat.m

- **See also**

[epeeg2spm](#) ^[26]

gdf2eeg

- **Description**

Conversion tool for GDF (">General Data Format for Biosignals ^[59]) files to EEG format.

- **Usage**

gdf2eeg mygdffile myparfile myeegfile

with :

- mygdffile : GDF file to convert (with extension).
- myparfile : text file containing conversion parameters (electrode names and numbers).
- myeegfile : output EEG file (no extension).

- **Fields of parameter file and example**

electrodes	Output channel name list :
P1.21	
P2.22	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
H1.1091	
H2.1092	
A1.225	The order should be the same as in the input GDF file.
A2.226	
Status.-1	

- **Example**

- **Comments**

- **Current version**

1.01 20-03-2009

- **History**

- 1.00 15-10-2008 (PEA) : first version.
- 1.01 20-03-2009 (PEA) : minor modification.

- **Files**

\$ELANPATH/bin/gdf2eeg
\$ELANPATH/bin64/gdf2eeg

- **See also**

igor2eeg

- **Description**

Conversion tool for IGOR (Waves) files to EEG format.

- **Usage**

igor2eeg myparfile myeegfile

with :

- myparfile : text file containing conversion parameters.
- myeegfile : output EEG file (no extension).

- **Fields of parameter file and example**

sampling_frequency 1000	Sampling frequency (Hz) of the output file.
channel_nb 6	Number of channels in output file (one Igor file = one channel).
channel_filename_list lfp_l1.ibw lfp_l2.ibw lfp_l3.ibw lfp_r1.ibw lfp_r2.ibw lfp_r3.ibw	Igor (Waves) file name list (with extension).
channel_name_list L1.1256 L2.1257 L3.1258 R1.2097 R2.2098 R3.2099	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
stim_nb 3	Number of files containing stimulations (events). If omitted, no event are imported.
stim_filename_list TimeTable_LP_REM.ibw TimeTable_LP_SWS.ibw TimeTable_LP_W.ibw	Igor (Waves) file name list (with extension) containing stimulations. Required if field "stim_nb" is present.
stim_code_list 1 2 3	Event code list to associate to each stimulation file above. Required if field "stim_nb" is present.

- **Example**

- **Comments**

- One Igor file contains only one LFP channel.

- **Current version**

1.03 19-04-2016

- **History**

- 1.00 28-07-2008 (PEA) : 1st version.
- 1.01 29-07-2008 (PEA) : support double and float data types. Support event list.
- 1.02 04-12-2015 (PEA) : Igor Wave file v5 fixes (64 bits issues).
- 1.03 19-04-2016 (PEA) : fix overflow error in data conversion (physical maximum estimation error).

- **Files**

\$ELANPATH/bin/igor2eeg
\$ELANPATH/bin64/igor2eeg

- **See also**

instep2epeg

- **Description**

Conversion tool for InstEP format (version 4) to EEG format.

- **Usage**

instep2eeg

- **Interactive input**

Name of the InstEP file to convert :
myfile_in
Generic filename for the Elan output files (without extension) :
myfile_out

- **Example**

- **Comments**

- **Current version**

1.4 13-08-2007

- **History**

- 1.14 13-08-2007 (PEA) : minor modification.

- **Files**

\$ELANPATH/bin/instep2epeg

- **See also**

mat2eeg

- **Description**

Conversion function for Matlab file to EEG format.

- **Usage**

mat2eeg(m_data, filename, m_events, str_ori_file1, str_ori_file2, s_fs, v_label, v_channel_type, v_channel_unit, varargin)
with :

- m_data : data matrix (1 channel per line, 1 sample per column).
- filename : output filename (with extension) with the complete path.
- m_events : table with the events (samples and event codes).
- str_ori_file1 : acquisition system.
- str_ori_file2 : acquisition place.
- s_fs : sampling frequency (Hz).
- v_label : name of the channels (with numbers in elec.dat).
- v_channel_type : sensor types.
- v_channel_unit : sensor units.
- option :
'Channel_Filter', v_channel_filter : filter description for each channel.

- **Fields of parameter file and example**

- **Example**

Load data with :

```
[m_data, m_events, v_label_selected, s_fs, s_nb_samples_all, s_nb_channel_all, v_label_all, v_channel_type_all, v_channel_unit_all, str_ori_file1, str_ori_file2] = eeg2mat('my_data.eeg', s_sample_start, s_sample_stop, v_channel_list)
```

Modify your data as you want, and create new matrix m_data_new.

Save new data in file with :

```
mat2eeg(m_data_new, 'my_new_data.eeg', m_events, str_ori_file1, str_ori_file2, s_fs, v_label_selected, v_channel_type_all(v_channel_list), v_channel_unit_all(v_channel_list))
```

- **Comments**

- Note to INSERM U821 users : ActiCap electrode positions can be loaded from file [ActiCap_elec.mat](#) ^[42].

- **Current version**

1.07 10-02-2012

- **History**

- 1.04 09-08-2010 (PEA) : minor change.
- 1.05 20-04-2011 (PEA) : sets Terminator value to 10 to have the Linux line return (when running in Windows) and fixes digital channels definition in header file.
- 1.06 31-05-2011 (PEA) : Fixes initialization of data_out to add the 2 digital channels (didn't work with only one channel).
- 1.07 10-02-2012 (PEA) : Fixes error printing message (with printf instead of disp).

- **Files**

\$ELANPATH/misc/matlab/mat2eeg.m

- *See also*

[eeg2mat](#) ^[1]

Attachment	Size
ActiCap_Elec.mat ^[51]	428 bytes

mat2ep

- *Description*

Conversion function Matlab file to Evoked Potential format.

- *Usage*

`n = mat2ep(p_file_name, s_Event_Code, s_fs, s_Nb_Sample_PreStim, v_Elec, s_Nb_Event_Aver, DATA)`
with :

- `p_file_name` : output *.p file name.
- `s_Event_Code` : averaged event code.
- `s_fs` : sampling frequency (Hz).
- `s_Nb_Sample_PreStim` : number of samples for prestimulus.
- `v_Elec` : channel number in elec.dat.
- `s_Nb_Event_Aver` : number of averaged epochs.
- `DATA` : data (1 channel per line, 1 sample per column).

output :

- `n` : number of written float.

- *Fields of parameter file and example*

- *Example*

Load data with :

```
[HEADER1, HEADER2, DATA] = ep2mat('my_data.p')
```

Modify your data as you want, and create new matrix DATA_NEW with same size.

Save new data in file with :

```
n = mat2ep('my_new_data.p', HEADER1.s_Event_Code, 1000/HEADER2.s_Sampling_Period,
HEADER2.s_Nb_Sample_PreStim, HEADER2.v_Elec, HEADER2.s_Nb_Event_Aver, DATA_NEW)
```

- *Comments*

- *Current version*

1.03 09-08-2010

- *History*

- 1.03 09-08-2010 (PEA) : minor change (variable names).

- *Files*

\$ELANPATH/misc/matlab/mat2ep.m

- *See also*

[ep2mat](#) ^[3]

mat2tf

- *Description*

Conversion function Matlab file to Time-Frequency format.

- *Usage*

```
mat2tf(tf_file_name, s_type_tf, s_type_1_tf, s_type_2_tf, s_type_signal, m_Data_TF, v_Elec,
v_Channels_Label, s_Nb_Sample_PreStim, s_Sampling_Frequency, s_Wavelet_Type, v_Frequency_Bins,
v_Wavelet_m_parameter, s_Wavelet_Window_Size, s_DownSampling_Step, v_Nb_Evt_per_Chan, s_corr_BL,
s_smooth_tf, s_Avg_Chan)
```

with :

- `tf_file_name` : output file name.
- `s_type_tf` : file type :
 - 1 : elementary time-frequency (avg.tf).

- 2 : computed from evoked potentials (ep.tf).
- 3 : phase (pl.tf).
- 6 : statistics (stat.tf).
- 10 : synchrony (.ph.tf) .
- s_type_1_tfmo : file origin :
 - 1 : no specific operation.
 - 2 : mean of TF.
 - 3 : median of TF.
 - 4 : difference between TF.
 - 5 : weighted mean of TF.
 - 6 : sum of TF.
 - 7 : Z of TF.
 - 8 : P of TF.
 - 9 : synchrony of TF.
- s_type_2_tfmo : if statistics, statistics type :
 - 1 : no statistic.
 - 2 : F Quade.
 - 3 : T Conover.
 - 4 : Z sign.
 - 5 : P sign.
 - 6 : n + sign.
 - 7 : n - sign.
 - 8 : Z Wilcoxon.
 - 9 : standard deviation.
 - 10 : upper confidence bound.
 - 11 : lower confidence bound.
 - 12 : synchrony randomization.
 - 13 : amplitude randomization.
- s_type_signal : original signal type :
 - 1 : potential.
 - 2 : SCD.
- m_Data_TF : time frequency data (s_Nb_Channels x s_Nb_Frequencies x s_Nb_Samples).
- v_Elec : channel indices in elec.dat .
- v_Channels_Label : channel labels.
- s_Nb_Sample_PreStim : number of samples for prestimulus.
- s_Sampling_Frequency : sampling frequency (Hz).
- s_Wavelet_Type : wavelet type (1 : Morlet, 2 : Gabor).
- v_Frequency_Bins : frequency bins values (Hz).
- v_Wavelet_m_parameter : m parameter for Morlet wavelet, or sigma t for Gabor wavelet.
- s_Blackman_Window_Size : number of samples of Blackman window :
 - default : number of samples of Blackman window (tf_nb_sample_blackman).
 - smooth : 0.
 - Wilcoxon : $tf_nb_sample_blackman * Te_Orig_sig(ms) / tfstat_time_step (ms)$.
- s_DownSampling_Step : downsampling step.
- v_Nb_Evt_per_Chan : number of averaged TF per channel.
- s_corr_BL : type of baseline correction :
 - 0 : none.
 - 1 : subtract mean of baseline.
 - 101 : subtract mean of baseline (computed on another file).
 - 2 : subtract median of baseline.
 - 102 : subtract median of baseline (computed on another file).
 - 3 : normalize and subtract mean of baseline.
 - 103 : normalize and subtract mean of baseline (computed on another file).
 - 4 : normalize and subtract median of baseline.
 - 104 : normalize and subtract median of baseline (computed on another file).
 - 5 : normalize and subtract RMS value.
 - 105 : normalize and subtract RMS value (computed on another file).
 - 6 : normalize and subtract maximum value.
 - 106 : normalize and subtract maximum value (computed on another file).
- s_smooth_tf : smoothing TF flag :
 - 0 : none.
 - 1 : smoothing.
- s_Avg_Chan : grand-average across-channels flag :
 - 0 : none.
 - 1 : grand-average.

- **Fields of parameter file and example**

- **Example**

- **Comments**

- **Current version**

1.01 09-08-2010

- **History**

- 1.00 02-11-2009 (PEA/EM) : first version.
- 1.01 09-08-2010 (PEA) : minor change (variable names).

- **Files**

\$ELANPATH/misc/matlab/mat2tf.m

- **See also**

tf2mat

micromed2eeg

- **Description**

Conversion tool for Micromed TRC files (versions 1, 2, 3, 4) to EEG format.

- **Usage**

micromed2eeg mymicromedfile.trc myeegfile [myparfile]

with :

- mymicromedfile.trc : Micromed file (with extension).
- myeegfile : output EEG file (no extension).
- option :
myparfile : text file containing conversion parameters. If omitted, it runs as if all fields of parameter are omitted.

- **Fields of parameter file and example**

electrodes P1.21 P2.22 H1.1091 H2.1092 A1.225 A2.226 Cable.-1	Output channel name list : Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat. The order should be the same as in the input Micromed file. This field is optional . If omitted, the name of the channels is read from the Micromed file. The first entry corresponding to the label in elec.dat is choosen. A file <i>myeegfile.elec.out</i> is created and contains the channel names and channel numbers used for the conversion.
conversion 1 1 1 1 1 0	Flag for conversion of electrodes : 1 : convert electrode, 0 : electrode not present in the output file (myeegfile) This field is optional . If omitted, all electrodes are converted.
notes_code 99	Event code number written in the myeegfile.eeg for each note (related to a function key) stored during the acquisition with the Micromed system. The file <i>myeegfile.notes.txt</i> is created. It contains the event positions (in samples from the beginning of the recording) and the comments associated with each function key. This field is optional . If omitted, no note is converted and no myeegfile.notes.txt is created.
event_chan_nb 2	Number of channels in which analog trigger pulses were recorded (superimposed to the EEG signal). If a trigger is found (saturation value), an event is detected and the value is replace by an interpolated value. This field is optional but required for analog trigger pulse detection. If omitted, no trigger pulse is detected.
event_chan_list 1 2	List of the channel numbers in which analog trigger pulses were recorded (superimposed to the EEG signal). This field is required when the field "event_chan_nb" is present.
event_code_list 32 64	List of event codes corresponding to the trigger pulses in which analog trigger pulses were recorded (superimposed to the EEG signal). Required for analog trigger pulse detection. If two or more trigger pulses occur simultaneously, the corresponding event code will be the sum of the concerned event codes. This field is required when the field "event_chan_nb" is present.

- **Example**

- **Comments**

- Trigger codes could be stored in the myMicromedfile.trc, either as digital events (trc file version 4), or as analog trigger pulses superimposed on selected EEG channels (see fields event_chan_nb, event_chan_list and event_code_list above).

- **Current version**

1.26 07-05-2014

- **History**

- 1.13 06-01-2006 (PEA) : bug fix.
- 1.14 01-02-2006 (PEA) : minor modification.
- 1.15 08-03-2006 (PEA) : adding field "conversion" for choosing channels to convert.
- 1.16 13-02-2007 (PEA) : suppress blank in electrode names.
- 1.17 21-02-2007 (PEA) : minor modification : display channels used for analog events.

- 1.18 26-02-2007 (PEA) : correction for event detection on analog channels and code notes for Micromed files version 3 and 4.
- 1.19 13-06-2007 (PEA) : test memory available before allocation.
- 1.20 13-08-2007 (PEA) : minor modification.
- 1.21 17-10-2007 (PEA) : adding chronological sort of digital triggers before use. Bad order may be an acquisition error : display a warning.
- 1.22 16-09-2010 (PEA) : dynamic allocation for reading elec.dat (no maximum anymore).
- 1.23 16-09-2010 (PEA) : minor modification.
- 1.24 17-03-2011 (PEA) : adds use of multiplier of sampling frequency as defined by Micromed.
- 1.26 07-05-2014 (PEA) : adds display to console of hipass and lopass filtering values read from TRC.

- **Files**

\$ELANPATH/bin/micromed2eeg

- **See also**

[micromed2ep](#) ^[52]

micromed2ep

- **Description**

Conversion tool for Micromed TRC files (versions 1, 2, 3, 4) to EP format.

- **Usage**

```
micromed2ep myMicromedEPfile.trc [myEPfile [myparfile]] [-v]
```

with :

- myMicromedEPfile.trc : The Micromed EP filename to be converted (with extension).
- options:
 - myEPfile : the name of the converted EP file (without extension). If not specified, the Micromed file is not converted and its content is displayed.
 - myparfile : conversion parameter file. If omitted, conversion parameters are extracted from Micromed TRC file.
 - v : verbose mode on (default is off).

- **Fields of parameter file and example**

electrodes

P1.21

P2.22

H1.1091

H2.1092

A1.225

A2.226

Cable.-1

Output channel name list :

Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.

The order should be the same as in the input Micromed file.

This field is **optional**.

If omitted, the name of the channels is read from the Micromed file. The first entry corresponding to the label in elec.dat is chosen.

- **Example**

- **Comments**

- Trigger codes could be stored in the myMicromedfile.trc, either as digital events (trc file version 4), or as analog trigger pulses superimposed on selected EEG channels (see fields event_chan_nb, event_chan_list and event_code_list above).

- **Current version**

1.00 15-04-2014

- **History**

- 1.00 15-04-2014 (PEA) : 1st version.

- **Files**

\$ELANPATH/bin/micromed2ep

- **See also**

[micromed2eeg](#) ^[14]

neuromag2eeg

- **Description**

Conversion tool for Neuromag/Elekta FIFF continuous files to Elan EEG format.

- **Usage**

neuromag2eeg myNeuromagfile.fiff [myeegfile myparfile]

with :

- myNeuromagfile.fiff : Neuromag/Elekta file to convert (with extension).
- options (if omitted, the FIFF file is read, and informations about channels are displayed) :
 - myeegfile : output EEG file (no extension).
 - myparfile : text file containing conversion parameters (sensor names and numbers).

- **Fields of parameter file and example**

sensors	Output channel name list :
M0113NM.5419	
M0112NM.5420	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if
M0111NM.5421	the channel is not defined in elec.dat .
M0122NM.5422	The order should be the same as in the input Neuromag file.
M0123NM.5423	
M0121NM.5424	
M0132NM.5425	
M0133NM.5426	
M0131NM.5427	
...	

- **Example**

Examples of conversion files :

- conversion parameter file for 306 MEG channels, 9 stimulations channels, 60 EEG channels and 1 EOG channel : [neuromag_376chan.conv.par](#) ^[53]
- conversion parameter file for 306 MEG channels, 9 stimulations channels, 2 EOG channels and 1 MISC channel: [neuromag_318chan.conv.par](#) ^[53]

- **Comments**

- The Neuromag system with 122 channels have 2 planar gradiometers at the each sensor location. The Neuromag system with 306 channels have 2 planar gradiometers and 1 axial magnetometer at the each sensor location. When converted, it is not possible to draw a topographical map of all channels. You must select only one of the 2 or 3 sensors of each location.

- **Current version**

1.01 04-05-2011

- **History**

- 1.00 08-09-2010 (PEA) : 1st version
- 1.01 04-05-2011 (PEA) : adds more precision (10 decimals) in for saving sampling period in output header file.

- **Files**

\$ELANPATH/bin/neuromag2eeg
\$ELANPATH/bin64/neuromag2eeg

- **See also**

Attachment	Size
neuromag_318chan.conv.par ^[54]	4.01 KB
neuromag_376chan.conv_par ^[55]	4.63 KB

neuroscan2eeg

- **Description**

Conversion tool for Neuroscan continuous files to EEG format.

- **Usage**


```
neuroscan2eeg [-?] myNeuroscanfile.cnt [myparfile myeegfile [calibration_factor] [-b32]]
```

with :

- myNeuroscanfile.cnt : Neuroscan file to convert (with extension).
- myparfile : text file containing conversion parameters (electrode names and numbers).
- myeegfile : output EEG file (no extension).
- options :
 - ? : display the structure of the parameter file.
 - calibration_factor : forces the amplitude calibration factor to be the same for all channels with this specified value. If omitted, calibration factors stored in the Neuroscan file (*.cnt) will be considered.
 - b32 : data are considered as 32 bit signed integers instead of 16 bit signed integers.

If only myNeuroscanfile.cnt is passed, the program reads the header of the file, prints it and stops.

- **Fields of parameter file and example**

electrodes	Output channel name list :
P1.21	
P2.22	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
H1.1091	
H2.1092	
A1.225	The order should be the same as in the input Neuroscan file.
A2.226	
Cable.-1	

- **Example**

- **Comments**

- **Current version**

1.15 22-10-2013

- **History**

- 1.13 13-08-2007 (PEA)
- 1.14 17-11-2011 (PEA) : adds the ability to read neuroscan header by giving the Neuroscan file name only.
- 1.15 22-10-2013 (PEA) : fixes an error in 64 bits version when reading file (use of long for 32 bits int).

- **Files**

\$ELANPATH/bin/neuroscan2eeg

- **See also**

pos2ctfmrk

- **Description**

Conversion tool for Elan event files (.pos) to CTF dataset (MarkerFile.mrk).

- **Usage**

```
pos2ctfmrk myelanfile.pos myfile.conv.par myctfdirectory.ds [MarkerFile.mrk.save]
```

with :

- myelanfile.pos : name of the file containing the events (with extension).
- myfile.conv.par : name of the file containing conversion parameters. It is created by ctfmrk2pos and can be modified. Be careful of CTF specifications when modifying this file/
- myctfdirectory.ds : CTF dataset directory (with extension) where the file MarkerFile.mrk will be written. Warning : the previous file will be overwritten.
- option :
 - MarkerFile.mrk.save : name of the saved MarkerFile.mrk (without directory.ds)(if omitted, the original MarkerFile.mrk will be replaced by the new one).

- **Fields of parameter file and example**

- **Example**

- **Comments**

- **Current version**

1.06 23-11-2010

- **History**

- 1.00 11-01-2006 (PEA) : first version.
- 1.01 14-03-2006 (PEA) : modification for new conversion file created by ctfmrk2pos. It allows to link events (Elan .pos) to class names (CTF).
- 1.02 16-03-2006 (PEA) : adding choice to save the original MarkerFile.mrk file.
- 1.03 17-03-2006 (PEA) : bug fix for number of characters per line. Runs without problem from this version.
- 1.04 13-08-2007 (PEA) : minor modification.
- 1.05 10-09-2009 (PEA) : correction of type definition in MegDefs.h for 64 bits compilation.
- 1.06 23-11-2010 (PEA) : correction in rchmpl (addition of maximum length line) in libgen.

- **Files**

\$ELANPATH/bin/pos2ctfmrk
\$ELANPATH/bin64/pos2ctfmrk

- **See also**

ctfmrk2pos

spike2_2eeg

- **Description**

Conversion tool for Spike2 files (.son, .smr) to ELAN EEG format.

- **Usage**

spike2_2eeg infile.smr [outfile.eeg file.par [+v]]

with :

- infile.smr : Spike2 file to convert (with extension).
If no output and parameter file is specified, the Spike2 file is read and info about it are displayed. No conversion is done.
- outfile.eeg : output EEG file (with extension).
- file.par : conversion parameter file (with extension). It provides channel names and numbers in elec.dat
- option :
+v : verbose mode on (default is off).

- **Fields of parameter file and example**

electrodes	Output channel name list :
LFP1.-1	
LFP2.-1	Names and numbers corresponding to elec.dat definition for converted channels, or name and -1 if the channel is not defined in elec.dat.
event1.-1	
	The order should be the same as in the input Spike2 file.

- **Example**

- **Comments**

1. ADC, event and real wave channels (kind 1, 2, 3, 4, 9) are converted to ELAN data channels.
2. Marker, ADC marker, real marker and text marker channels (kind 5, 6, 7, 8) are converted to ELAN events.
3. WARNING : channel kind 2, 3, 4 (event channels), 6 (ADC marker), 7 (real marker), 8 (text marker) and 9 (real wave channel) have not been tested. Please provide us a file to test it if you want to convert this kind of data (see [Contact us](#) ^[56] section).

- **Current version**

1.03 19-03-2012

- **History**

- 1.00 20-05-2011 (PEA) : 1st version.
- 1.01 12-03-2012 (PEA) : minor modification (checks for positive block size before reading).
- 1.02 13-03-2012 (PEA) : minor modification (displays an error and exits when a problem occurs reading data header blocks).
- 1.03 19-03-2012 (PEA) : adds partial support to SON library V9 (unsupported feature : LUTable).

- **Files**

\$ELANPATH/bin/spike2_2eeg

- *See also*

tf2mat

- *Description*

Conversion function for Elan TF files (*.tf*) to Matlab.

- *Usage*

[s_Nb_Frequencies, v_Frequency_Bins, s_Nb_Samples, s_Nb_Channels, s_Sampling_Frequency, s_Nb_Sample_PreStim, s_Wavelet_Type, s_Blackman_Window_Size, s_DownSampling_Step, v_Channels_Label, v_Elec, v_Wavelet_m_parameter, v_Nb_Evt_per_Chan, m_Data_TF] =

tf2mat(tf_file_name,varargin)

with :

- tf_file_name : input filename (with extension) with the complete path.
- option :
 - 'save', mat_name : to save all the results in a .mat file named mat_name.mat .

output :

- s_Nb_Channels : total number of channels.
- v_Elec : channel number in elec.dat.
- v_Channels_Label : channel label.
- s_Nb_Samples : total number of samples in time domain.
- s_Nb_Sample_PreStim : number of samples for prestimulus.
- s_Sampling_Frequency : sampling frequency (Hz).
- s_Nb_Frequencies : total number of frequencies.
- s_Wavelet_Type : wavelet type (1 : Morlet, 2 : Gabor).
- v_Frequency_Bins : frequency bins values (Hz).
- v_Wavelet_m_parameter : m parameter for the Morlet wavelet, or sigma t parameter for Gabor wavelet.
- s_Blackman_Window_Size : number of samples of the Blackman window.
- s_DownSampling_Step : downsampling step.
- v_Nb_Evt_per_Chan : number of averaged TF per channel.
- m_Data_TF : time-frequency data (s_Nb_Channels x s_Nb_Frequencies x s_Nb_Samples).

- *Fields of parameter file and example*

- *Example*

- *Comments*

- *Current version*

1.01 09-08-2010

- *History*

- 1.00 31-10-2009 (PEA/EM) : first version.
- 1.01 09-08-2010 (PEA) : minor change (variable names).

- *Files*

\$ELANPATH/misc/matlab/tf2mat.m

- *See also*

mat2tf

Lyon Neuroscience Research Center - Brain Dynamic and Cognition team

CRNL



Source URL: http://elan.lyon.inserm.fr/?q=ref_conv_tool

Links:

- [1] <http://elan.lyon.inserm.fr/?q=eeg2mat>
- [2] <http://elan.lyon.inserm.fr/?q=mat2eeg>

[3] <http://elan.lyon.inserm.fr/?q=ep2mat>
[4] <http://elan.lyon.inserm.fr/?q=mat2ep>
[5] <http://elan.lyon.inserm.fr/?q=tf2mat>
[6] <http://elan.lyon.inserm.fr/?q=mat2tf>
[7] http://elan.lyon.inserm.fr/?q=ref_event_proc_tool
[8] http://elan.lyon.inserm.fr/?q=s1_GS
[9] <http://elan.lyon.inserm.fr/?q=biosemi2eeg>
[10] <http://elan.lyon.inserm.fr/?q=brainamp2eeg>
[11] <http://elan.lyon.inserm.fr/?q=ctf2eeg>
[12] <http://elan.lyon.inserm.fr/?q=EGIraw2eeg>
[13] <http://elan.lyon.inserm.fr/?q=instep2epeg>
[14] <http://elan.lyon.inserm.fr/?q=micromed2eeg>
[15] <http://elan.lyon.inserm.fr/?q=neuroscan2eeg>
[16] <http://elan.lyon.inserm.fr/?q=neuromag2eeg>
[17] <http://elan.lyon.inserm.fr/?q=alphamap2eeg>
[18] <http://elan.lyon.inserm.fr/?q=ascii2eeg>
[19] <http://elan.lyon.inserm.fr/?q=edf2eeg>
[20] <http://elan.lyon.inserm.fr/?q=gdf2eeg>
[21] <http://elan.lyon.inserm.fr/?q=igor2eeg>
[22] <http://elan.lyon.inserm.fr/?q=ctf2ep>
[23] <http://elan.lyon.inserm.fr/?q=ctfssv2ep>
[24] <http://elan.lyon.inserm.fr/?q=ep2fieldtrip>
[25] <http://elan.lyon.inserm.fr/?q=ep2nutmeg>
[26] <http://elan.lyon.inserm.fr/?q=ep2spm>
[27] <http://elan.lyon.inserm.fr/?q=ep2mex>
[28] <http://elan.lyon.inserm.fr/?q=ctfmrk2pos>
[29] <http://elan.lyon.inserm.fr/?q=pos2ctfmrk>
[30] <http://elan.lyon.inserm.fr/?q=sites/default/files/conv-GSN-128.par>
[31] <http://elan.lyon.inserm.fr/sites/default/files/conv-GSN-128.par>
[32] <http://elan.lyon.inserm.fr/sites/default/files/sgn200.par>
[33] <http://elan.lyon.inserm.fr/?q=EGIav2ep>
[34] <http://sccn.ucsd.edu/eeglab/>
[35] <http://biosig.sourceforge.net/>
[36] <http://elan.lyon.inserm.fr/?q=eegpos>
[37] <http://elan.lyon.inserm.fr/?q=sites/default/files/elecActiCap.dat>
[38] <http://elan.lyon.inserm.fr/sites/default/files/elecActiCap.dat>
[39] http://elan.lyon.inserm.fr/sites/default/files/exemple_bti.coil
[40] http://elan.lyon.inserm.fr/sites/default/files/exemple_ctf.coil
[41] http://elan.lyon.inserm.fr/sites/default/files/exemple_Neuromag.coil
[42] <http://elan.lyon.inserm.fr/?q=book/export/html/18>
[43] <http://elan.lyon.inserm.fr/?q=sites/default/files/ctf275.par>
[44] <http://elan.lyon.inserm.fr/sites/default/files/ctf275.par>
[45] <http://www.edfplus.info/>
[46] <http://fieldtrip.fcdonders.nl/>
[47] <http://elan.lyon.inserm.fr/?q=eeg2fieldtrip>
[48] <http://nutmeg.berkeley.edu/>
[49] <http://www.fil.ion.ucl.ac.uk/spm/>
[50] <http://elan.lyon.inserm.fr/?q=Scientific>
[51] http://elan.lyon.inserm.fr/sites/default/files/ActiCap_Elec.mat
[52] <http://elan.lyon.inserm.fr/?q=micromed2ep>
[53] <http://elan.lyon.inserm.fr/?q=sites>
[54] http://elan.lyon.inserm.fr/sites/default/files/neuromag_318chan.conv_.par
[55] http://elan.lyon.inserm.fr/sites/default/files/neuromag_376chan.conv_.par
[56] <http://elan.lyon.inserm.fr/?q=contact>