

# ***Tcl Batch Commands***

## ***User's Guide***

### ***Compumedics Neuroscan***

7850 Paseo Del Norte  
El Paso, TX 79912  
USA

Voice :915-845-5600  
Fax: 915-845-2965  
e-mail: [sales@neuro.com](mailto:sales@neuro.com)  
[techsup@neuro.com](mailto:techsup@neuro.com)  
web site: [www.neuro.com](http://www.neuro.com)



**For Technical Support with BATCH files, we strongly recommend you send your questions via e-mail to [techsup@neuro.com](mailto:techsup@neuro.com), and that you include your batch file as an attachment to the e-mail. This will greatly facilitate our troubleshooting, it will speed the response, and it will give you an electronic copy to refer to. Tech Support will help with problems specific to SCAN commands, however, you should refer to a Tcl text for more complex issues regarding Tcl commands and functionality. Please see our batch web site for additional information and support options (<http://www.neuro.com>).**

**Copyright © 2003 - Compumedics Neuroscan**

All rights reserved. Printed in the United States of America. No part of this manual may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the company. Making copies of any part of this document for any purpose other than your own personal use is a violation of United States copyright laws. For information, contact Neuroscan.

***Compumedics Neuroscan***

7850 Paseo Del Norte  
El Paso, TX 79912  
Telephone: 915-845-5600  
Fax: 915-845-2965  
E-mail: [techsup@neuro.com](mailto:techsup@neuro.com)  
Web: [www.neuro.com](http://www.neuro.com)

## *Table of Contents*

BATCH Files Using Tool Command Language.....	9
Executing BATCH Files in SCAN.....	9
Tcl Batch Command Structure.....	11
Tcl BATCH Command Conventions.....	11
Standard Conventions.....	12
Zero-Based Index.....	15
Quotes versus Braces.....	15
Creating Batch Files.....	16
Auto-write Feature.....	18
Conditional and Looping Commands.....	18
Summary of Some Flow Control Commands.....	19
The EXPR Command and Some Arithmetic Operators.....	20
Get and Set Commands.....	21
Variable Substitution.....	22
File Naming Strategies.....	24
Manipulating File Names: the GLOB Command.....	25
Summary of Related Commands.....	26
Using Autorun.....	27
Sample Batch Files.....	28
Executing BATCH Files Outside of SCAN.....	30
Tcl Batch Commands.....	30
Tcl Commands for ACQUIRE and EDIT.....	31
<b>ADDDISPLAYFILTER/REMOVEDISPLAYFILTER</b> .....	31
<b>ADDWAVEBOARD</b> .....	32
<b>ARRANGEWINDOWS</b> .....	32
<b>BUTTERFLYPLOT</b> .....	33
<b>CHANLIST</b> .....	33
Data Scaling Commands.....	34
<b>AUTOSCALE</b> .....	34
<b>SCALE</b> .....	35
<b>SCALEX</b> .....	35
File Management Utility Commands.....	35
<b>DELETE</b> .....	35
<b>NEWNAME</b> .....	36
<b>RENAME</b> .....	36
GET/SET Commands.....	36
<b>GETBASELINECOR/SETBASELINECOR</b> .....	36
<b>GETCHANATTRIBUTE</b> .....	37
<b>SETCHANATTRIBUTE</b> .....	37
<b>GETCHANLABEL</b> .....	38
<b>GETCHANNELINDEX</b> .....	38
<b>GETDISPLAYINVERTED/SETDISPLAYINVERTED</b> ..	39
<b>GETDISPLAYPAGE/SETDISPLAYPAGE</b> .....	39
<b>GETINPUT</b> .....	40

GETINPUTFILE.....	41
GETNUMCHANS.....	41
GETNUMDISPLAYPAGES.....	42
GETNUMPOINTS.....	42
GETNUMSWEEPS.....	43
GETOUTPUTFILE.....	43
GETSWDISPLAYSPEED/SETSWDISPLAYSPEED.....	44
GETSWEEPMAX.....	44
GETSWEEPMIN.....	45
HERTZTOPOINT.....	45
INSTRUCT.....	46
ISOFFLINE/ISONLINE.....	47
LATENCYTOPOINT.....	47
MAXIMIZE.....	48
MINIMIZE.....	48
PAUSE.....	48
"PICKOFF" Commands.....	48
PICKOFFDIRECTORY.....	48
PICKOFFEXTENSION.....	48
PICKOFFFILENAME.....	48
PICKOFFNAMEONLY.....	49
POINTTOHERTZ.....	49
POINTTOLATENCY.....	50
PRINTWAVEBOARD.....	50
READPOS/SAVEPOS.....	50
REM.....	51
RESTORE.....	51
REVIEW.....	51
SETSPECTDISPLAYSTYLE.....	52
Subject Information Commands.....	52
READSUB.....	52
SAVESUB.....	52
SUBJECT.....	52
Substitutions.....	52
INSERT.....	53
WRITELOG.....	53
Zoom Commands.....	54
TOGGLEZOOM.....	54
ZOOMIN.....	54
ZOOMOUT.....	54
Discontinued DOS Commands for ACQUIRE and EDIT.....	54
Tcl Commands for ACQUIRE.....	55
CALIB.....	55
CLEARAVG.....	55
CONFIGUREBLINKNOISEREDUCTION.....	55
CONFIGUREEKGNOISEREDUCTION.....	56

<b>DODCCORRECT</b> .....	56
<b>DOIMPEDANCE</b> .....	57
<b>ENABLEBLINKNOISEREDUCTION</b> .....	57
<b>ENABLEEKGNOISEREDUCTION</b> .....	57
<b>GETAST</b> .....	58
<b>HIGHIMPEDANCE</b> .....	58
<b>PAUSEDISPLAY</b> .....	59
<b>RESTART</b> .....	59
<b>STARTACQUISITION</b> .....	59
<b>STARTRECORDING</b> .....	60
<b>STOPACQUISITION</b> .....	60
<b>STOPRECORDING</b> .....	60
Various Acquisition Script Examples.....	60
<b>VIEW</b> .....	62
Discontinued DOS Commands for ACQUIRE.....	62
 Tcl Commands for EDIT.....	63
<b>Auto-write Feature</b> .....	63
<b>"_EX" Commands</b> .....	65
<b>ADD</b> .....	65
<b>ADDANNOTATION</b> .....	66
Appending Data Files.....	66
<b>APPEND</b> .....	66
<b>APPENDRECORDING</b> .....	67
<b>CONCATCNT</b> .....	67
<b>MERGE</b> .....	67
<b>AREAREPORT</b> .....	68
Artifact Correction/Rejection.....	68
<b>ARTCOR</b> .....	69
<b>ARTREJ</b> .....	69
<b>ARTREJ_EX</b> .....	70
<b>ARTREJCNT</b> .....	70
<b>ARTREJCNTFREQ</b> .....	71
<b>ARTREJFREQ</b> .....	71
<b>ARTREJFREQ_EX</b> .....	72
<b>BLINKNOISEREDUCTION</b> .....	72
<b>CLEARART</b> .....	73
<b>EKGNOISEREDUCTION</b> .....	73
<b>READREJ</b> .....	74
<b>SAVEREJ</b> .....	75
Averaging.....	75
<b>AVERAGE</b> .....	75
<b>AVERAGE_EX</b> .....	76
<b>GROUPAVG</b> .....	76
<b>AVGBANDS</b> .....	77
Baseline Correction.....	77

BASECOR.....	78
BASECOR_EX.....	78
EXCLUDEFORBASECOR.....	79
RESETFORBASECOR.....	79
CLEARALLMARKERS.....	79
CLEARMARKERS.....	79
CLOSEALL.....	79
CLOSEFILE.....	80
COHERENCE.....	80
SETCOHREF.....	81
COHERENCE_EX.....	81
CREATESORT/DELETESORT.....	81
CUTEPOCH.....	83
DCCORRECT.....	83
DECIMATE.....	84
DELETEBADCHANNELS.....	84
DELETESWEEPS.....	84
Detrending (Linear Detrending).....	85
DETREND.....	85
DETREND_EX.....	85
EXCLUDEFORDETREND.....	85
RESETFORDETREND.....	86
ENABLEOVERWRITEPROMPT.....	86
EPOCH.....	86
EPOCH_EX.....	87
ERBP.....	87
ERCOH.....	88
Exporting/Importing ASCII Files.....	89
EXPORTAVG.....	89
EXPORTAVG_EX.....	90
EXPORTCNT.....	91
EXPORTCNT_EX.....	91
EXPORTCOH_EX.....	92
EXPORTEDF.....	92
EXPORTEEG.....	92
EXPORTEEG_EX.....	93
IMPORTAVG.....	93
IMPORTCNT.....	94
IMPORTEEG.....	94
EXTRACT.....	95
Filtering Data Files.....	95
FILTER.....	95
FILTER_EX.....	95
EXCLUDEFORFILTER.....	96
RESETFORFILTER.....	96
FSP.....	96

<b>FSP_EX</b> .....	97
GET/SET Commands.....	97
<b>GETCOMP/DELETEALLCOMP</b> .....	98
<b>GETCURRENTEPOCHINDEX/</b> <b>SETCURRENTEPOCHINDEX</b> .....	98
<b>GETEPOCHCOUNT</b> .....	98
<b>GETEPOCHINFO/SETEPOCHINFO</b> .....	99
<b>GETEVENTCOUNT</b> .....	100
<b>GETEVENTINFO/SETEVENTINFO</b> .....	101
<b>GETMARKERCOUNT</b> .....	103
<b>GETMARKERINFO</b> .....	104
<b>GETPOINTDATA</b> .....	105
<b>GFP</b> .....	106
<b>INSERTSTIMEVENT</b> .....	106
<b>INSERTRESPONSEEVENT</b> .....	107
<b>INTEGRATEWAVEFORM</b> .....	108
<b>LDR</b> .....	108
<b>LOAD3DD</b> .....	108
<b>MERGEevt</b> .....	109
Open/Save Commands.....	109
<b>OPENFILE</b> .....	109
<b>SAVEAS</b> .....	109
Peak Detection.....	110
<b>EXCLUDEFORPEAKDETECTION</b> .....	110
<b>PEAKDETECTION</b> .....	110
<b>PEAKDETECTION_EX</b> .....	110
<b>RESETFORPEAKDETECTION</b> .....	111
<b>SAVEPEAK</b> .....	111
<b>PRINT</b> .....	112
<b>RECTIFY</b> .....	113
<b>RECTIFY_EX</b> .....	113
<b>REFER</b> .....	113
<b>REJECTBLOCK</b> .....	114
<b>RESPWIN</b> .....	114
<b>SAVEEVENT</b> .....	115
<b>SAVEEVENT_EX</b> .....	115
<b>SELECTFILE</b> .....	116
Smoothing Data.....	116
<b>EXCLUDEFORSMOOTH</b> .....	116
<b>RESETFORSMOOTH</b> .....	117
<b>SMOOTH</b> .....	117
<b>SMOOTH_EX</b> .....	117
<b>SNR</b> .....	117
<b>SORT</b> .....	118
<b>SPATIALFILTER</b> .....	118
<b>SPATIALSVD</b> .....	119

Spectral Analysis.....	120
<b>FORWARDFFT/INVERSEFFT</b> .....	120
<b>SPECTRUM</b> .....	121
<b>SPLINEFIT</b> .....	121
Statistical Comparisons.....	122
Correlational Measures.....	122
<b>CROSSCORR</b> .....	122
<b>CROSSCORR_EX</b> .....	122
<b>INTRACLASSCORR</b> .....	123
<b>INTRACLASSCORR_EX</b> .....	123
<b>PEARSONS</b> .....	123
<b>PEARSONS_EX</b> .....	124
T-test Comparisons.....	124
<b>PAIRED</b> .....	124
<b>TSCORE</b> .....	125
<b>ZSCORE</b> .....	125
<b>SUBTRACT</b> .....	126
<b>UPDATECHANGES</b> .....	126
<b>VOLTAGETHRESHOLD</b> .....	126
Windowing Data Files.....	127
<b>EXCLUDEFORWINDOWDATA</b> .....	127
<b>RESETFORWINDOWDATA</b> .....	127
<b>WINDOWDATA</b> .....	127
<b>WINDOWDATA_EX</b> .....	128
<b>WRITENUMSWEEPS</b> .....	128
Discontinued DOS Commands for EDIT.....	128
 Advanced Tcl Scripts.....	130
Complete Batch Sequence.....	130
Advanced Artifact Rejection Application.....	131
Changing Information in the Event Table.....	132
Using the Same Code Repeatedly.....	134
Commands for ACQUIRE and EDIT (alphabetical list).....	135
Commands for ACQUIRE (alphabetical list).....	136
Commands for EDIT (alphabetical list).....	136
Discontinued DOS Commands (alphabetical list).....	139
<b>Quick Reference Guide</b> for TCL Commands.....	142



## BATCH Files Using Tool Command Language

BATCH commands and BATCH command files are used for automating procedures in SCAN. BATCH reads an ASCII file containing a series of commands and processes these commands in the order that they are received. Writing a BATCH command file is similar to writing a computer program in that it involves thinking out command steps in an exact sequence.

There are distinct advantages to using BATCH files. For one, they spare you the tedium of performing the same analysis operations across files, while at the same time they ensure that the exact same analysis steps will be followed across data files. Additionally, they provide a log of the analysis operations that were performed, the parameters used by each, and the order in which they were applied. You may find it useful to save a copy of the BATCH file along with archived data files.

In SCAN 4.3 and in subsequent versions, you have the option to create BATCH files outside of SCAN, and then run them in SCAN. These can be used in place of the Scripting options in EDIT, if desired. The BATCH files use Tool Command Language, and have .Tcl extensions (and are usually referred to as "tickle" files). The commands are similar to those used in earlier SCAN 3.0 and 4.0 BATCH files (.CMD files), in that they are created in a text editor, and each line consists of a command followed by a string of arguments (or parameters), and/or path and file name designators.

Your BATCH programs can be as simple as a list of the basic commands found in this document, or as complicated as Tool Command Language will allow. If you are doing fairly simple, straightforward BATCH processing, you may find that the Scripting options in EDIT are perfectly adequate and more user-friendly. Examples of more complex programs are presented in various parts of the manual (see, for example, the sections on Variable Substitution and GET/SET commands, and the Advanced examples at the end of the manual). For more complex programming, you should buy a Tcl textbook. A good one is: Tcl and the Tk Toolkit by John K. Ousterhout (the Tcl author). Please see <http://www.neuro.com/neuroscan/batch.htm> for a link to a site to order the book, as well as for additional support options

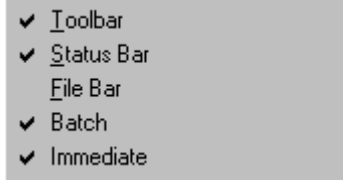
## Executing BATCH Files in SCAN






When you enter EDIT or ACQUIRE, you will see the BATCH and Immediate Toolbars.




The pull-down arrow will display the last 10 commands that were executed, for your convenience. The Toolbars can be repositioned at the top or bottom of the display, as desired, by dragging and dropping them, as with most Toolbars.

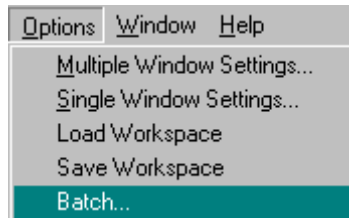
If you do not see the BATCH controls, or if you close one or both of the Toolbars, go to View on the Main Menu bar and enable the Batch or Immediate options.



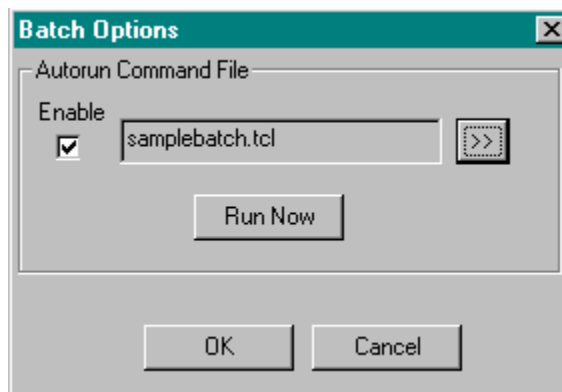
*Batch* - The BATCH controls are used to run the Tcl files you have created. Use the Browse button  as needed to select the Tcl file you wish to run. The Batch Editor is a text editor designed especially for Tcl BATCH files . After retrieving a .tcl file, the  button (Run) will become active. If you are using the PAUSE or REVIEW commands, the program will stop at that point until you click the  button (Resume). The  button (Stop) will terminate the BATCH file.

*Immediate* - The Immediate Mode line allows you to enter and execute a single line BATCH command. Enter the line and click the  button (Run) to execute it. The pull-down arrow allows you to select from previous commands you have executed.

*Autorun.* Another BATCH option is accessed from the Options / Batch option.



Selecting this option displays the Autorun feature controls.

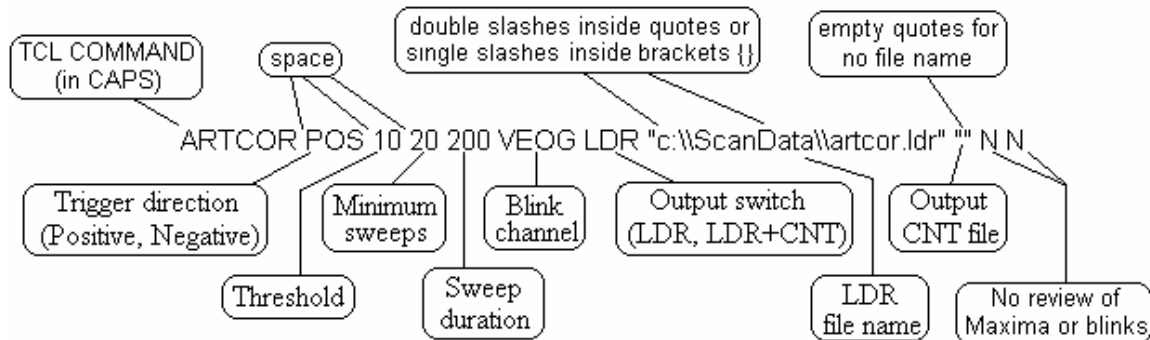


The Autorun feature allows you to create a library of smaller programs (similar in function to subroutines) that may be accessed from other programs that you run,

or from the Immediate command line. Operational details are described in the *Using Autorun* section below.

## Tcl BATCH Command Structure

Let's look first at a typical command. Below is how the command for applying the ocular artifact reduction transform might appear.



The Tcl command appears first (always in CAPS with no spaces). Following that are the arguments, or parameters. These are listed for each command in the pages below. The arguments are separated by one or more spaces. Quotes "" or braces {} are used to define the beginning and ending of text strings (double slashes \\ must be used inside of quotes, and single slashes \ must be used inside braces). Empty text strings also use quotes (or braces), as shown above at the end of the line.

The example above displays the basic command line. A simple BATCH file is shown below. This retrieves an epoched file, filters it, performs a linear detrend transform, and then averages the sweeps.

```
OPENFILE "c:\\ScanData\\p300.eeg"
FILTER BANDPASS ZERO 1.5 24 50 24 0 0 0 0
"c:\\ScanData\\filtp300.eeg"
OPENFILE "c:\\ScanData\\filtp300.eeg"
DETREND ENT 0 0 "c:\\ScanData\\detrendp300.eeg"
OPENFILE "c:\\ScanData\\detrendp300.eeg"
AVERAGE T N N "" 0 0 0 "" "c:\\ScanData\\averagep300.avg"
CLOSEALL
```

## Tcl BATCH Command Conventions

There are some standard conventions used with Tcl. These are somewhat different from other programming languages, so you are encouraged to read them even if you are familiar with programming techniques. The basic conventions are summarized below.

## Standard Conventions

There are some standard conventions used by Tcl files.

**\$.** The dollar sign causes Tcl to perform a variable substitution; the dollar sign and the variable following it are replaced with the value of the variable you define. For example, you can replace a frequently used path, such as, `c:\ScanData`, with a variable you create and substitute: `$path`. See the **Variable Substitutions** section below.

**Braces{ }.** Braces are used to indicate text strings (including empty ones), as in the following example `{c:\ScanData\myfile.cnt}`. Note that single slashes are used in the path. Conversely, text strings may be enclosed by quotes, as in the following example `"c:\ScanData\myfile.cnt"`. Note that double slashes are used within the quotes. Braces **MUST** be used in commands that contain a List variable type. Braces are also used as the beginning and ending designators for commands in a loop or subroutine (see the **Quotes versus Braces** section below).

**Capitals.** Commands must be written in CAPITAL letters, such as `OPENFILE` `"c:\ScanData\myfile.cnt"`.

**Command.** The beginning of each line is the command, such as `OPENFILE`, `EPOCH`, `FILTER`, etc. After the command, skip at least one space, and enter a value for EVERY parameter (use `""` or `{ }` for empty text strings), unless otherwise indicated..

**Comments.** This is used to place informational text in the file that has no effect on the actual BATCH program. Comments are designated by placing `"#"` or `REM` at the beginning of a line. If you want to add a comment after the Tcl command and parameters, on the same line, leave at least one space, and then type `;"#"`, followed by your comments. For example:

```
OPENFILE {c:\ScanData\test.eeg}    ;# open the file to be processed
```

**Conditional and Looping commands.** You may control the flow of the BATCH program using conditional and looping commands. These are explained in the **Conditional and Looping Commands** section below.

**Double Slashes and Quotes.** Double slashes must be used when defining paths, as in the example `"c:\ScanData\myfile.cnt"`, *when the path is enclosed with quotes* (see also, Single Slashes and Braces). Exceptions are the `CREATESORT` and `DELETESORT` commands, which use no quotes or braces.

**Parameters.** Following the command in a line in the Tcl file come the values for the parameters. Each command has its own parameters, and a value must be entered into each one, even if that value is a 0 or "". Spaces must separate the parameters.

**Quotes** " ". Quotes are used to indicate text strings (including empty ones), as in the following example "c:\\ScanData\\myfile.cnt". Note that double slashes are used in the path. Conversely, text strings may be enclosed by braces, as in the following example: {c:\\ScanData\\myfile.cnt}. Note that single slashes are used within the braces.

**Semicolons/Multiple Commands per Line.** You may have more than one command on a line. Leave at least one space at the end of one command, then place a semicolon, and write the next command.

**Single Slashes and Braces.** Single slashes must be used when defining paths, as in the example {c:\\ScanData\\myfile.cnt}, *when the path is enclosed with braces* (see also, Double Slashes and Quotes).

**Spaces.** Spaces are used to separate the command from the parameters, as well as between each of the parameters on a command line.

**Substitutions.** Tcl allows you to perform several kinds of substitutions, including substituting values for variables and for commands. Details for using substitutions may be found in the **Variable Substitution** section below, and in more detail in a Tcl text.

**Variable Types.** There are several different types of variables: Boolean, defined value, double, float, int, list, and string.

**Boolean.** A Boolean parameter can have only two states: 0=false, and non-0=true. The Boolean variable type will recognize equivalent entries. For example, you may enter Off, No, n, 0, or False for a parameter, and they will all be interpreted the same way. Similarly, On, Yes, y, 1 or True may be used in the parameter field interchangeably. The entries are not case sensitive (NO = no = No, etc.).

**defined value.** Defined values are used where there are several options from which to choose, such as, SUM, MEAN, and AREA. Type in the entire word, or as much of the word as is necessary to insure that it will be distinguished from the other possible responses. For example, you could type S, M or A in the above example, since each letter is unique. In the case of BANDPASS and BANDSTOP, you would need to enter either BANDP or BANDS to insure a unique interpretation. It is recommended that you enter enough of the word to make it easy to recognize in the

command line. (It might also avoid confusion in the future if more options are added to the particular parameter).

**double.** Double is, or can be, a very large signed floating point number (5000, -200.932, 0.5), and may have a decimal point.

**float.** Float is a signed floating point number. Floating point means that this variable can contain a decimal point (1.0, 2.5, 3.1415).

**int.** Int is a signed integer, like 12, -3, and 0, with no decimal points (whole numbers).

**list.** A List is a series of elements, separated by spaces, and enclosed by braces. For example, a list might be {Sterling Virginia El Paso Texas}. All five words are treated as single elements. The elements can be combined by using pairs of braces within the outer pair: {{Sterling Virginia} {El Paso Texas}}. These are treated as two elements. The same result can be accomplished with quotes inside the outer braces: {"Sterling Virginia" "El Paso Texas"}.

In the SCAN BATCH commands, you will find some that contain lists of electrode channels or data files. Placing the channels/files in a List avoids having to repeat the same command for each channel/file. For individual electrodes, or other elements without backslashes, the Tcl command will appear something like (see the electrode list):

```
APPEND {c:\ScanData\secondfile.avg} {P3 P4 C3 C4}
      {c:\ScanData\finalfile.avg}
```

For lists of files, where there are backslashes in the paths, each element must be enclosed in braces, and the entire list must also be enclosed in braces.

```
DELETE {{c:\ScanData\file1.avg} {c:\ScanData\file2.avg}
      {c:\ScanData\file3.avg}}
```

The line can also be written with quotes within braces, as follows:

```
DELETE {"c:\\ScanData\\file1.avg" "c:\\ScanData\\file2.avg"
      "c:\\ScanData\\file3.avg"}
```

If all elements are to be included, you may simply use ALL rather than listing them individually:

```
DELETE {{ALL}}.
```

A related command is **LAPPEND**. This command lets you add elements to a list interactively (see the Ousterhout Tcl text, page 65).

*Note that you may not use variable substitutions within braces.* You must use braces with the list parameter. Therefore, to use variable substitutions with lists, use a format similar to the following:

```
MERGE [list $path\file1.eeg $path\file2.eeg $path\file3.eeg]
$path\outputfile.eeg
```

**string.** String is a text string of variable length, such as a file name, a path and a file name, or sorting criteria {1,2, 3-6}. Strings are enclosed by quotes or braces. Empty strings, "" or {}, must be used as place holders if they are not used in the command (with some exceptions, as noted).

Null strings ("" ) can be used with all commands that require files including ldr files, dat files, data in, data out, event, etc., and the file is required. In these cases, "" will bring up the Open File or Save As utility, so you don't have to input the file name and path. These commands are noted in the on the File name parameter lines as: File name (or ""). There are exceptions to this function; these will also be noted by the message ("" will fail).

## Zero-Based Index

Several of the BATCH commands use a zero-based index (a common programming convention). This means that when items are counted, the first one is always zero, not one. For example, if you use the **GETCURRENTEPOCH-INDEX** command (returns the sweep number of the displayed EEG file), the returned value will be one less than the value displayed in the Status box in EDIT. That is because the BATCH command assumes, in this case, that the first sweep is sweep zero, not sweep one.

## Quotes versus Braces

In the Convention information presented thus far, you may get the impression that quotes and braces can be used interchangeably. In fact, there are two important differences between them. First, *braces nest*. That is, you may have a structure such as {a{b{c}}}} with braces, whereas you cannot nest quotes within quotes. Second, *no substitutions can occur inside braces*, whereas they occur inside double quotes.

Braces are used for more than just specifying text strings. Braces (not quotes) are used in defining loops (or what can be thought of as subroutines). The example below shows a loop within a loop. Note that in this example, as well as others contained in this manual, a single brace may be isolated on a line, or on a line by itself:


```
for {set index 0} {$index < $evcount} {incr index} {
    set type [GETEVENTINFO $index -EventType]
    if {$type == 0} {
        set stimcode [GETEVENTINFO $index -StimulusCode]
        set response [GETEVENTINFO $index -KeypadCode]
        SETEVENTINFO $index -Stim [expr {$stim+$value}]
    }
}
```


This is a standard programming convention employed to make it easier to track loops and nested loops. You typically indent commands in a loop, and you indent further for nested loops. You could write the commands as follows (also eliminating the indents), and there would be no difference in how the script is executed.

```
for {set index 0} {$index < $evcount} {incr index}
{set type [GETEVENTINFO $index -EventType]
if {$type == 0}
{set stimcode [GETEVENTINFO $index -StimulusCode]
set response [GETEVENTINFO $index -KeypadCode]
SETEVENTINFO $index -Stim [expr {$stim+$value}]}}
```

Spacing the braces and using indentations makes it easier to see what the script is doing (and also to make sure there is the same number of open and closed braces).

## Creating BATCH Files

The Tcl Batch Editor  (new in SCAN 4.3) provides the easiest means for creating or modifying BATCH files, and it is strongly recommended over other text editors (such as Word, Wordpad and Notepad). The Tcl BATCH Editor was designed especially for Tcl

BATCH files. Click the  icon to open the program. It is very easier to use, and most of its functionality is obvious and self-explanatory. One thing you will notice immediately is that the text appears in colors.

Comments are in green, Tcl commands are in blue, text strings are in red, and so forth. This is purely for your

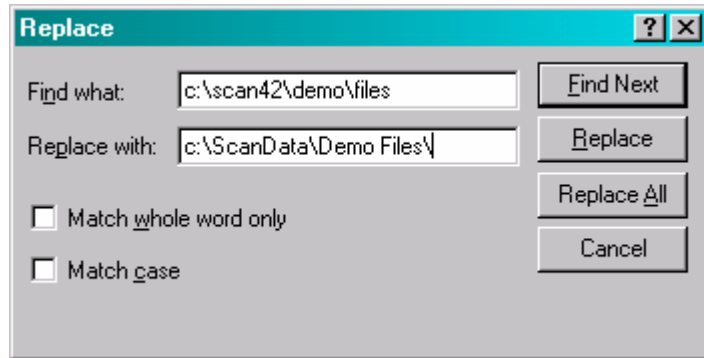
```
# change the path as needed
ENABLEOVERWRITEPROMPT N
set path "c:\scan43 test files\pearsons"

OPENFILE "$path\cor_1.avg"
INSTRUCT "Follow the P&C test procedures for Time Domain Test-1."
PEARSONS_EX {c:\scan43 test files\Pearsons\cor_2.avg} -25 99.5 {c:\sr
CLOSEALL
```

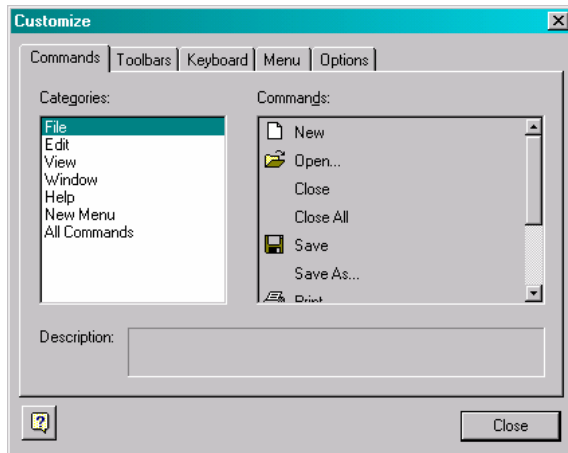


convenience. The remainder of the functionality is for the most part the same as with any other text editor. You may, for example, copy text from a different source, such as the Immediate or History Toolbars, or from an e-mail, other Windows application, or PDF file, and Paste it into the editor. The colors are added automatically.

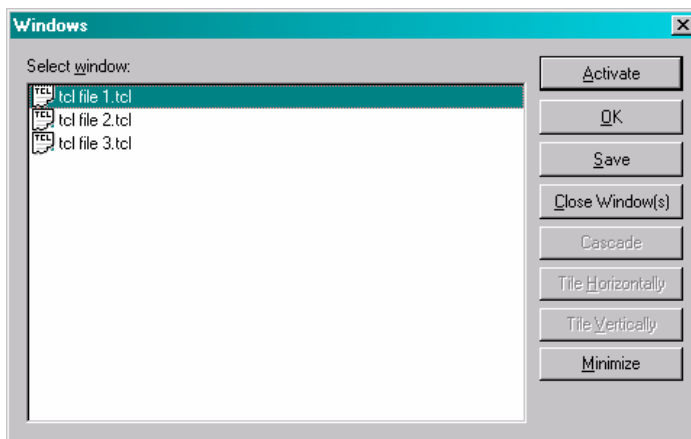
There are some options that you will find especially useful, or that may not be familiar to you. Once have a file open (New or existing), the Edit option appears. Under it is the Replace option. This displays the Replace dialog display. Use it to search and replace sections in the BATCH file. For example, if you need to change the paths in your batch files, you can do it easily.



You have flexibility in the appearance of the Toolbar in the Editor. Under View, click Toolbar, and then Customize to see the following display. The tabs contain various options for modifying the Toolbars, creating Shortcut keys, and various additional options.




Another useful option is Windows, under the Window command on the Main Menu bar. Clicking it displays the following dialog screen (three BATCH files are open).



This display lets you select from the open files, close them individually, save them individually, and so forth. (Highlight more than one file to activate the Cascade and Tile options). The Windows screen is another convenience for managing multiple open files.

In practice, the BATCH files are typically left open as you debug the BATCH file. Just Save the file, leave the editor open, and select and then run the file from the BATCH toolbar in EDIT. Once you have

In practice, the BATCH files are created in the editor, and it is

selected the file in EDIT, you did not need to reselect the same file. Make any changes in the editor, save the file, and click the  button to run the revised BATCH file.

## Auto-write Feature

Beginning with SCAN version 4.3, we have added an "auto-write" feature in the EDIT program. Whenever you execute a transform in point-and-click (P&C) mode, the Tcl BATCH command will be created automatically, and displayed in the History toolbar field. Most of the commands also appear in the Immediate field. From either place you can copy the line into the Tcl BATCH Editor in the batch file you are creating. See the example at the beginning of the "Tcl Commands for EDIT" section below for more details.

## Conditional and Looping Commands

Conditional statements and looping commands are integral parts of Tcl. There are several related commands, including **if**, **for**, and **foreach**, to name some of the more commonly used ones. (Please refer to a Tcl text for complete details).

**if.** The **if** command evaluates an expression, test its results, and conditionally executes a script based on the result. An example is abstracted from the longer BATCH file example in the **GET and SET Commands** section below.

```
set path [GETINPUTFILE "select file" "cnt" "c:\\ScanData\\Batch Examples\\"]  
if {path == "Cancel"} {return}
```

In this example, the **if** commands tests whether the variable "path" equals the returned string "Cancel". If it does, the program is terminated. If not, the program continues on the next line (not shown in the example).

Two of the more common commands for looping are **for** and **foreach**. Each of these executes a script over and over - they differ in the kinds of setup they do prior to each iteration and in the ways they terminate the loop. Additional flow control commands are summarized in the next section.

**for.** The **for** command takes four arguments: the first is an initialization script, the second is an expression that determines when to terminate the loop, the third is the reinitialization script (which is evaluated after each execution of the loop before evaluating the test again), and the fourth argument is the script that forms the body of the loop. An example is abstracted from the sample BATCH file in the **GET and SET Commands** section below.

*Note: The open brace "{" must appear at the end of the **for**, **foreach** and **if** lines. This differs from other programming languages, and is a requirement for TCL.*

```

set valuetoadd 10
for {set index 0} {$index < $evcount} {incr index} {
    set type [GETEVENTINFO $index -EventType]
    if {$type == 0} {
        set stimcode [GETEVENTINFO $index -StimulusCode]
        set response [GETEVENTINFO $index -KeypadCode]
        SETEVENTINFO $index -StimulusCode [expr
            {$stimcode+$valuetoadd}]
    }
}

```

In the example, the first argument sets the "index" variable to zero. The second is the "test" that is performed. The third argument increments the "index" variable by 1 for each loop, and the fourth argument is the actual script.

**foreach.** The **foreach** command iterates over all of the elements in a list. It contains three arguments: the first is the name of a variable, the second is a list, and the third is the body of the script file. The example below is abstracted from the sample BATCH file in the **Substitutions** section below.

```

set path "c:\\ScanData\\"
ENABLEOVERWRITEPROMPT F
set filelist {file1.avg file2.avg file3.avg}
foreach element $filelist {
    set file $path$element
    OPENFILE "$file"
    BASECOR PRE 0 0 N N
    SAVEAS "$path$element"
}

```

The variable **filelist** contains a list of data files. The **foreach** command applies the script to each element in the file list. The **ENABLEOVERWRITEPROMPT F** command allows the existing files to be overwritten automatically without user intervention.

**Summary of Some Flow Control Commands.** The following list summarizes some of the flow control commands you may find useful (abstracted from Tcl and the Tk Toolkit, JK Ousterhout, 1994; question marks denote optional arguments):

<b>break</b>	Terminates the innermost nested looping command.
<b>continue</b>	Terminates the current iteration of the innermost looping command and goes on to the next iteration of that command.
<b>for init test reinit body</b>	Executes <b>init</b> as a Tcl script. Then evaluates <b>test</b> as an expression. If it evaluates to non-zero then

executes **body** as a Tcl script, executes **reinit** as a Tcl script, and reevaluates **test** as an expression. Repeats until **test** evaluates to zero. Returns an empty string.

**foreach** **varName** **list** **body** For each element of **list**, in order, set variable **varName** to that value and execute **body** as a Tcl script. Returns an empty string. **list** must be a valid Tcl list.

**if** **test1** **body1** **?elseif** **test2** **body2** **elseif** ...? **?else** **bodyn**?  
Evaluates **test1** as an expression. If its value is non-zero, executes **body1** as a Tcl script and returns its value. Otherwise evaluates **test2** as an expression; if its value is non-zero, executes **body2** as a script and returns its value. If no test succeeds, executes **bodyn** as a Tcl script and returns its result.

**source** **fileName** Reads the file whose name is **fileName** and evaluates its contents as a Tcl script. Returns the results of the script. The sourced file can be another Tcl BATCH file, which can be run within the first BATCH file (similar to a subroutine).

**switch** **?options?** **string** **pattern** **body** **?pattern** **body** ...?  
**switch** **?options?** **string** {**pattern** **body** **?pattern** **body** ...?}  
Matches **string** against each **pattern** in order until a match is found., then executes the **body** corresponding to the matching **pattern**. If the last **pattern** is **default**, it matches anything. Returns the results of the **body** executed, or an empty string if no pattern matches. **options** may be **-exact**, **-glob**, **regexp**, or **--** to indicate the end of options.

**while** **test** **body** Evaluates **test** as an expression. If its value is non-zero, executes **body** as a Tcl script and reevaluates **test**. Repeats until **test** evaluates to zero. Returns an empty string.

**The EXPR Command and Some Arithmetic Operators.** In the BATCH examples presented, you will encounter the **EXPR** command as well as some basic arithmetic operators. These are summarized for your convenience (refer to a Tcl text for more details; question marks denote optional arguments):

**EXPR** **arg** **?arg** **arg** ...? - Concatenates all the **arg** values together (with spaces between), evaluates the results as an expression, and returns a

string corresponding to the expression's value. For example, [expr \$offset+100] adds 100 to the variable \$offset.

The following are commonly used arithmetic operators (abstracted from Tcl and the Tk Toolkit, JK Ousterhout, 1994; in the Results column, 1 means true and 0 means false).

Syntax	Result	Operand types
-a	Negative of a	int, real
!a	Logical NOT: 1 if a is zero, 0 otherwise	int, real
a*b	Multiply a and b	int
a/b	Divide a by b	int, real
a+b	Add a and b	int, real
a-b	Subtract b from a	int, real
a<b	1 if a is less than b, 0 otherwise	int, real, string
a>b	1 if a is greater than b, 0 otherwise	int, real, string
a<=b	1 if a is less than or equal to b, 0 otherwise	int, real, string
a>=b	1 if a is greater than or equal to b, 0 otherwise	int, real, string
a==b	1 if a equals b, 0 otherwise	int, real, string
a!=b	1 if a is not equal to b, 0 otherwise	int, real, string
a&& b	Logical AND: 1 if a and b are non-zero, 0 otherwise	int, real
a  b	Logical OR: 1 if either a is non-zero or b is non-zero, 0 otherwise	int, real
a?b:c	Choice: if a is non-zero then b, else c	a: int, real

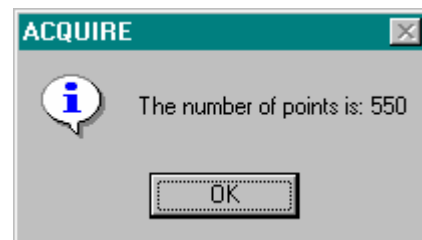
## Get and Set Commands

In SCAN 4.3 BATCH files you will find quite a few GET and SET commands. These are used to perform a variety of functions having to do with obtaining information about various data file parameters, or setting values in the file. Some of the more important commands are those that access the event table (where the stimulus and response events are stored), and then allow you to modify the information in the event table (see the **Advanced Tcl Scripts** section for an example).

The GET and SET commands may be used in several ways. Perhaps the simplest is to return information about the data file. For example, let's say you retrieve an EEG or AVG file, and wish to know how many data points there are in each sweep. The INSTRUCT command returns the information for simple viewing.

```
OPENFILE "c:\\ScanData\\vep.avg"
set points [GETNUMPOINTS]
INSTRUCT "The number of points is: $points"
```

In other cases, you may wish to return the information, and then apply it in subsequent commands. In the following example, the information returned from the



LATENCYTOPOINT commands is used in the GETPOINTDATA command.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vep.avg"
set start [LATENCYTOPOINT 97]
set stop [LATENCYTOPOINT 104]
set data [GETPOINTDATA $start $stop {OZ}]
INSTRUCT $data
CLOSEALL
```

## Variable Substitution

One of the most useful aspects of BATCH files is that you may perform the same sequence of operations on *multiple files*. Variable substitutions allow you to do this (as well as other useful operations). For those not familiar with programming in general, and with Tcl programming in particular, we will demonstrate the basic uses of variable substitutions, as you might likely employ them in SCAN BATCH files. (Refer to a Tcl text for more complete details). See also the INSERT command below for more basic applications.

The essential commands are the **set** command and the **\$** character. Here is a simple example.

```
set x 4
```

This sets the created variable **x** to contain the string **4**. A more likely example might be:

```
set file 001
```

The variable **file** now contains the string **001**. If you then use the expression **\$file** in a command, **001** will replace **\$file**. You can use more complex strings, such as file names. These should be placed within quotes:

```
set a "p300.eeg"
OPENFILE "c:\\ScanData\\Batch Examples\\$a"
```

You may also substitute a created variable for the path you are using. For example:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\p300.eeg"
```

You can use multiple variable substitutions in the same command.

```
set path "c:\\ScanData\\Batch Examples\\"
set a "p300.eeg"
OPENFILE "$path$a"
```

Once the new variables have been created, they may be used repeatedly within the same BATCH file.

```
set a "p300.eeg"
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\$a"
BASECOR PRE 0 0 N N "$path\\base$a"
```

This BASECOR command performs a baseline correction, and the output file uses the `$path` variable to avoid having to type in the entire path, and creates the basep300.eeg file.

To create a BATCH file that performs the same operations on a series of files, use one of the following sequences. In the first instance, let's say you have files with a common root name followed by sequential numbers, such as, file1.avg, file2.avg and file3.avg.

```
set path "c:\\ScanData\\Batch Examples\\"
for {set i 1} {$i < 4} {incr i} {
    set root file
    set ext base
    set username "$path$root$i.avg"
    OPENFILE $username
    BASECOR PRE 0 0 N N
    SAVEAS "$path$root$i$ext.avg"
}
```

In the BATCH file above, we set `path` to indicate the path to the location of the data files. On the next line, using the `for` command (see the **Conditional and Looping Commands** section above for more details), we set `i` to 1, and then increment `i` as long as `i` is less than 4. We then create a variable called `root`, that contains the root name of the data files, which in this case is "file" (as in file1.avg). We then create another variable called `ext`, which contains the string "base". That extension is added to the final output file name to avoid overwriting the original data file.

Adding the extension using a substitution avoids the extra space problem you would encounter, between `$i` and `base`, if you said `SAVEAS "$path$root$i base.avg"`. Other ways are demonstrated in the next section.

The Next line creates the `username` variable, which is comprised of the `path`, `root`, and number (i.e., c:\\ScanData\\ Batch Examples\\file1.avg). Now, the `OPENFILE` command is used to open the `$username` file, and the `BASECOR` operation is performed. In the case, the output file is ignored on the `BASECOR` line, and the additional `SAVEAS` command is needed. The output files are file1base.avg, file2base.avg, etc.

Note that the BATCH commands within the are enclosed in braces, and there are variable substitutions contained within the commands. You cannot, however, use variable substitutions within braces in other circumstances. In general, no substitutions can occur within braces. The information contained in the braces is interpreted literally. Let's look at the MERGE command for an illustration.

The MERGE command is used to join two or more epoched files (.eeg extension). You must open one file first (OPENFILE), then specify the files to be added to it, and the final output file. It uses a list for the first parameter. In the case without variable substitutions:

```
MERGE {{c:\ScanData\data\file1.eeg} {c:\ScanData\data\file2.eeg}
{c:\ScanData\data\fileN.eeg}} {c:\ScanData\data\outputfile.eeg}
```

Braces MUST be used with this command (not quotes). That will run with any problems; however, consider the same case where variable substitutions are present:

```
set path {c:\ScanData\data}
MERGE {{$path\file1.eeg} {$path\file2.eeg} {$path\fileN.eeg}}
{$path\outputfile.eeg}
```

That will NOT run, because the \$path\file1.eeg line is interpreted literally. The substitution is ignored. Trying to use MERGE [list {\$path\file1.eeg}]... will present the same problem, and will NOT be executed. The following form WILL run (by removing the braces):

```
MERGE [list $path\file1.eeg $path\file2.eeg $path\file3.eeg]
$path\outputfile.eeg
```

**File Naming Strategies.** In the above example, we presented one strategy for creating output files that avoids the "extra space" problem you may encounter when you use a substituted variable with additional text in the file name. The problem occurs when you have a substituted variable, such as \$filename, and you wish to append additional text to it, such as the word "filter". If the output file command reads SAVEAS "\$path\$filenamefilter.avg, for example, that will not run - \$filenamefilter is treated as a single substituted variable name. There must be a space between the two: \$filename filter. Or, you can set filter as its own variable: \$filter, as in the above example (where the command would read \$filename\$filter).

Probably the easiest way to avoid the extra space problem is to enclose the variable name within braces {}, as in the following example.

```
$directory${filename}extensionname.avg
```



Another strategy that can be applied to solve the problem uses the **format** command. An example is as follows:

```
set path "c:\\ScanData\\Batch Examples\\"
set name "Test"

set file [format "%s%s_f.avg" $path $name]
```

The **format** command is followed by a string (in quotes) that contains "%s" for each string variable that is to be substituted (other arguments are used for different kinds of variables, such as, %d is used to indicate an integer - refer to a Tcl text for more details), plus additional text preceded by an underscore. After the string are the variables to be substituted, in the order that they are to be substituted. The will result will be a file named c:\\ScanData\\Batch Examples\\test\_f.avg.

You might also use the "pickoff" commands, as in the following example. The "pickoff" commands extract sections of a string (see **Pickoff Commands** in the EDIT section below). In the example, note that %d was used on the format line to agree with the number (integer) that was entered:

```
set path "c:\\ScanData\\Batch Examples\\viscpt.cnt"
OPENFILE $path

set Filetag [GETINPUT "File Tag" "Enter a number to differentiate the
data file"]

set Directory [PICKOFFDIRECTORY $path]
set Filename [PICKOFFNAMEONLY $path]
set Ext [PICKOFFEXTENSION $path]

set OutFile [format "%s%s_%d%s" $Directory $Filename $Filetag
$Ext]
SAVEAS $OutFile

OPENFILE $OutFile
```

If you enter "123" in the GETINPUT screen, OutFile will be c:\\scan4.3\\Batch Examples\\viscpt\_123.cnt.

**Manipulating File Names: the GLOB Command.** The Tcl **glob** command takes one or more patterns as arguments and returns a list of all the file names that match the pattern(s). This can be very useful in SCAN in such instances where you need to retrieve several data files. For example, the **glob** command can be used with the **GROUPAVG** command to retrieve all AVG files in a specified folder.

```
GROUPAVG [glob c:/ScanData/demo/*.avg] GROUP N 1 Y Y
{c:\scan4.3\demo\average.avg}
```

All of the AVG files in the indicated folder will be used in the group average. The **glob** command can also be used with, for example, the **foreach** command to avoid having to type all the file names in the list:

```
set list [glob c:/ScanData/demo/*.avg]
foreach element $list {
    OPENFILE $element
    PAUSE 1000
    CLOSEALL
}
```

Continuing with ways to perform substitutions in the BATCH files, you may encounter instances in which your data files do not have a common root and sequential numbers, but rather there are just a series of files with unrelated names. These can still be called and used in a single BATCH file using the **foreach** command (see **Conditional and Looping Commands** above for more details).

```
set path "c:\\ScanData\\Batch Examples\\"
ENABLEOVERWRITEPROMPT F
set filelist {file1.avg file2.avg file3.avg}
foreach element $filelist {
    set file $path$element
    OPENFILE "$file"
    BASECOR PRE 0 0 N N
    SAVEAS "$path$element"
}
```

In this example, we created a variable called **filelist**, which contains the files we want to include. On the next line, the **foreach** command creates a new variable called **element**, for each file in **filelist**, as the files are processed. The next line creates a variable called **file**, which contains the path specified on the first line, and the file name. The various files are then opened, and baseline corrected. As above, the output file is ignored in this case on the **BASECOR** command, so the **SAVEAS** command is used.

**Summary of Related Commands.** The following are some of the more commonly used Tcl commands for manipulating variables (abstracted from Tcl and the Tk Toolkit, JK Ousterhout, 1994). Optional arguments are enclosed in question marks.

**append** varName value ?value ...? - Appends each of the value arguments to variable varName, in order. If varName does not exist, it is created with

an empty value before appending. The return value is the new value of **varName**.

**incr varName ?increment?** - Adds **increment** to the value of variable **varName**. **increment** and the old value of **varName** must both be integer strings (decimal, hexadecimal, or octal). If **increment** is omitted, it defaults to 1. The new value is stored in **varName** as a decimal string and returned as a result of the command.

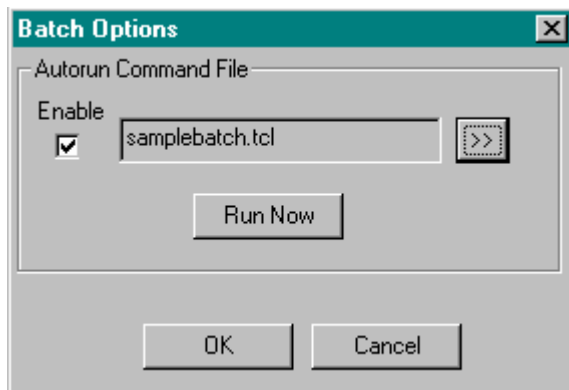
**set varName ?value?** - If **value** is specified, sets the value of variable **varName** to **value**. In any case the command returns the (new) value of the variable.

**unset varName ?varName varName ...?** - Deletes the variables given by the **varName** arguments. Returns an empty string.

## Using Autorun

Under Options / Batch..., you will see the Autorun feature.

The file that you specify will be executed automatically each time ACQUIRE or EDIT is run. It is intended to allow you to add custom commands. The autorun batch file should contain only procedures as mentioned in the Advanced Tcl Scripts section (Using the Same Code Repeatedly). Because these procedures are defined in a file that is executed at startup, all procedures will subsequently be available for use in BATCH files and in the immediate mode each time ACQUIRE or EDIT is run without any need for further consideration.



Example 1. Say the autorun batch has the following code:

```
proc OPEN { file } {  
    OPENFILE [format "%s%s" "c:\\ScanData\\Batch Examples\\" $file]  
}.  

```

Then the following command:

**OPEN viscpt.cnt**


can be used as a shortcut to replace the longer

OPENFILE "c:\\ScanData\\Batch Examples\\viscpt.cnt".

This is especially convenient for the Immediate mode.

Example 2. Procedures can be used to 'wrap' the embedded commands for easier access:



```
proc C {} {  
    CLOSEALL  
}
```

Now, simply typing C (note: these are case sensitive) in the Immediate window and pressing ENTER, or clicking , will close all opened data files.

Of course, these are very simple examples. These procedures can become very complex. One can imagine a procedure to calculate the difference waveform of a P300 recording that would reduce to:

```
proc DIFF {inputfile, resultfile} {  
    ...  
    ...(body omitted)  
    ...  
}
```

The body of the function could do all necessary epoching, filtering, sorting, averaging, and subtraction necessary to produce the desired result.

The  button is used to execute the specified file right away. If you make changes to your autorun file in a text editor and save them to disk, then pressing the  button will make the changes effective immediately in ACQUIRE or EDIT.

## Sample BATCH Files

There are a number of sample BATCH files included throughout this manual. They are presented generally to illustrate a specific command, however, in some cases they also illustrate the use of substitutions and various flow control commands. For example, the **ARRANGEWINDOWS** example also demonstrates the use of the "elseif" command. As you go through the manual, be sure to note the sample scripts, even if the specific command is one that you may not be using. Note also that there are several sample BATCH files in the \\Scan4.3\\Batch Examples folder. The "Example 8-I.tcl" file demonstrates the uses of several of the "GET" commands.

```
set path "c:\\ScanData\\Batch Examples\\"  
OPENFILE "$path\\100HzNeg"  
set channels [GETNUMCHANS]  
INSTRUCT "NUMBER OF CHANNELS: $channels"
```

```

set displaypages [GETNUMDISPLAYPAGES]
INSTRUCT "NUMBER OF DISPLAY PAGES: $displaypages"
set npoints [GETNUMPOINTS]
INSTRUCT "NUMBER OF POINTS PER CHANNEL: $npoints"
set sweepmax [GETSWEEPMAX]
INSTRUCT "MAXIMUM EPCOH LATENCY (ms): $sweepmax"
set sweepmin [GETSWEEPMIN]
INSTRUCT "MINIMUM EPOCH LATENCY (ms): $sweepmin"
set latencypoint [LATENCYTOPOINT 10]
INSTRUCT "DIGITIZATION POINT AT SPECIFIED LATENCY (ms):
    $latencypoint"
set nsweeps [GETEPOCHCOUNT]
INSTRUCT "NUMBER OF EPOCHS IN RECORDING: $nsweeps"

```

The next sample BATCH file retrieves the viscpt.cnt demo file, performs an ocular artifact reduction, and then epochs the file. The sweeps are averaged according to stimulus event type (see the **CREATESORT** command for details), and the resulting **FREQ** file is subtracted from the **RARE** file. **PZ** from the difference file is zoomed in on, and the figure is printed. Lastly, all files are closed.

```

OPENFILE {c:\ScanData\viscpt.cnt}
ARTCOR pos 10 20 400 "VEOG" LDR+ {c:\ScanData\viscpt.ldr}
    {c:\ScanData\viscptCORR.cnt} Y Y
OPENFILE {c:\ScanData\viscptCORR.cnt}
# Create the sort schemes
    CREATESORT RareSort
    RareSort -TypeEnabled yes -TypeCriteria "1"
    CREATESORT FreqSort
    FreqSort -TypeEnabled yes -TypeCriteria "2"
EPOCH port {} -100 700 n y n n NULL {c:\ScanData\viscpt.eeg}
OPENFILE {c:\ScanData\viscpt.eeg}
AVERAGE TIME n n "" 0 0 0 RareSort {c:\ScanData\RAREtype1.avg}
OPENFILE {c:\ScanData\RAREtype1.avg}
# The "focus" is now returned to the viscpt.eeg file
SELECTFILE {viscpt.eeg}
AVERAGE TIME n n "" 0 0 0 FreqSort {c:\ScanData\FREQtype2.avg}
# The "focus" is returned to the RAREtype1.avg file, so that the FREQtype2.avg
# file may be subtracted from it.
SELECTFILE {RAREtype1.avg}
SUBTRACT {c:\ScanData\FREQtype2.avg} {c:\ScanData\difference.avg}
OPENFILE {c:\ScanData\difference.avg}
ZOOMIN PZ
PRINT
CLOSEALL

```

*Note the use of **OPENFILE** to retrieve a newly created file. There are circumstances in which this may not be necessary, but the safest thing is to retrieve the file again (you will see a message if the file is already open). If not, you may find you are applying the subsequent operations to an unintended file.*

*Note also that virtually all of the sample BATCH files used in this manual are reproduced in the "Batch examples.doc" file, which is located in the Batch Examples folder. If you wish to run or modify any of the scripts, just create a tcl file by cutting and pasting sections from the document. (You may need to copy the sample data files that are used to the Batch Examples folder). The samples appear in the same order as in this manual. Samples not related to a specific BATCH command will be labeled using the following convention: Example 6-I is the 6th sample script in the Introduction section. Example 2-A is the second sample script in the Tcl Commands for Acquire section, and so on.*

For a more complex example of a BATCH file, with substitutions and loops, please see the **Advanced Tcl Scripts** section following the Tcl BATCH Commands (near the end of the manual).

## Executing BATCH Files Outside of SCAN

You may execute your Tcl BATCH files from the command line in Windows, or by including the command in an external batch file (.bat file), by using the following command:

```
Acquire.exe /EditMode /Batch=c:\ScanData\Batch Examples\sample.tcl.
```

The above command will run BATCH files for the offline EDIT part of the program. To run BATCH files for the online ACQUIRE part of the program, use:

```
Acquire.exe /Batch=c:\ScanData\Batch Examples\sample.tcl.
```

## Tcl BATCH Commands




---

The Tcl commands are listed below more or less in alphabetical order, based on the name of the operation as used in ACQUIRE or EDIT, or both. Related commands are placed together; for example, the various commands related to baseline correction are grouped under Baseline Correction. The commands are separated into those used in either the online ACQUIRE or offline EDIT modes, then those only for online ACQUIRE, followed by those for offline EDIT operations. SCAN 3 users will note that the STATS and WINDOW commands have been incorporated into EDIT. In many instances, you may need to refer to the same transform or operation in the ACQUIRE or EDIT manual for more details.

The actual commands used in the Tcl files are written with an **ARIAL** font. Typically, there are 3 columns below the command description. The first column is the parameter

number. The second column is the type of variable (int, boolean, defined value, double, float or string, described above). The last column is a description of the parameter.

Commands from the older DOS SCAN 3 CMD files are mentioned where relevant, although not all have been implemented. Many of the discontinued DOS commands have been superceded with newer commands, and others are simply obsolete. The older commands are included to help facilitate the rewriting of your older CMD files to Tcl files. Commands that have been superceded are noted as such, and you will be directed to the new command. New commands are denoted by **NEW**. Discontinued commands are in gray.

At the end of each Command name, you will see one or more symbols indicating the types of files for which the command may be applied. These are:  for CNT files,  for EEG files, and  for AVG files.

You may want to set your data files as "Read-only" to prevent them from being unintentionally overwritten. To do this, use the Windows Explorer to locate your data file. Right click on the file, and go to Properties. Set the Attributes for Read-only. (You can set the files automatically to be Read-only files when they are recorded; see the Misc tab option in ACQUIRE).



## Tcl Commands for ACQUIRE and EDIT

---

The following commands may be used in either the online ACQUIRE or offline EDIT programs.

**ADDDISPLAYFILTER/REMOVEDISPLAYFILTER** - Add/Remove Display Filters. These commands are used to add or remove filters to the active data file (this affects the display only - not the stored data). Filtering is FIR analog simulation (zero-phase not available). See also **FILTER**.

### ADDDISPLAYFILTER


1	defined value	Filter type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
2	float	High pass cutoff
3	int	High pass dBs (6, 12, 24, or 48)
4	float	Low pass cutoff
5	int	Low pass dBs (6, 12, 24, or 48)
6	float	Notch start frequency
7	float	Notch stop frequency
8	int	Notch poles
9	Boolean	Rectify
10	list	Channels to be filtered (or "All")

*Example.* ADDDISPLAYFILTER BANDP 0.5 24 30 24 0 0 0 NO ALL.  
The analog simulated filter with the selected parameters will be applied to the data being displayed (affects the display only).

## REMOVEDISPLAYFILTER

1 list Channels to remove display filter (or "All")

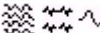
*Example.* REMOVEDISPLAYFILTER {FZ CZ PZ OZ}. The display filter will be removed from the indicated channels.

**ADDWAVEBOARD** - Add Waveform to Waveboard.  This command is used to add one, several, or all electrode channels to the Waveboard. The Waveboard display will appear, allowing you to modify and save the waveforms, as desired. The BATCH file will continue after the waveforms have been sent to the waveboard; use the PAUSE or REVIEW command to suspend the BATCH file if you want to modify or save the Waveboard waveforms at that point. (*Note: if the EDIT window is maximized to full size, the Waveboard window will be hidden behind it*).

1 list Electrode(s) to be sent to the Waveboard

*Example.* ADDWAVEBOARD {FZ CZ PZ OZ}. The indicated channels will be sent to the waveboard. A more complete example is as follows, where channels from two equivalent data files are sent to the Waveboard:

```
OPENFILE "c:\\ScanData\\Batch Examples\\sepbk.avg"
ADDWAVEBOARD {C3 C4}
OPENFILE "c:\\ScanData\\Batch Examples\\sepnobk.avg"
ADDWAVEBOARD {C3 C4}
# modify the files as desired
PAUSE      ;# if desired - the BATCH file will continue if there is no
PAUSE
# continue with any additional BATCH commands
```

**ARRANGEWINDOWS** - Arrange Windows in the Display.  When multiple displays are open, this command is used to arrange the displays using the standard Windows conventions.

1 defined value Method (CASCADE, TILEHORZ, TILEVERT)

*Example.* ARRANGEWINDOWS TILEV. This will arrange the windows in equal vertically oriented displays. The following example demonstrates ARRANGEWINDOWS.

```
set path "c:\\ScanData\\Batch Examples"
for {set n 0} {$n < 3} {incr n} {
    OPENFILE "$path\\sepbk.avg"
```



```

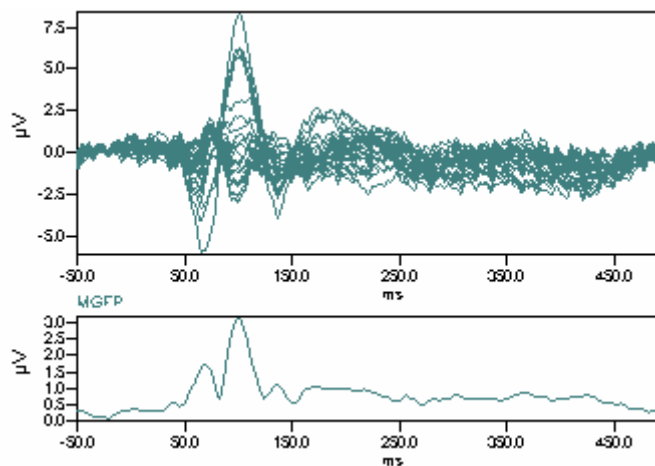
OPENFILE "$path\\sepnoblk.avg"
if {$n == 0} {
    ARRANGEWINDOWS CASCADE
    INSTRUCT "This is an example of Cascading Windows"
    CLOSEALL
}
elseif {$n == 1} {
    ARRANGEWINDOWS TILEHORZ
    INSTRUCT "This is an example of Horizontally Arranged
    Windows"
    CLOSEALL
}
elseif {$n == 2} {
    ARRANGEWINDOWS TILEVERT
    INSTRUCT "This is an example of Vertically Arranged
    Windows"
    CLOSEALL
}
}

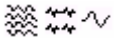
```

**BUTTERFLY PLOT** - Show as Butterfly Plot.  This command is used to display the time domain AVG and EEG files in the Butterfly Plot style.

1                      Boolean              Enable butterfly plot

*Example.* BUTTERFLY PLOT Y. The data will be displayed in the butterfly plot mode.

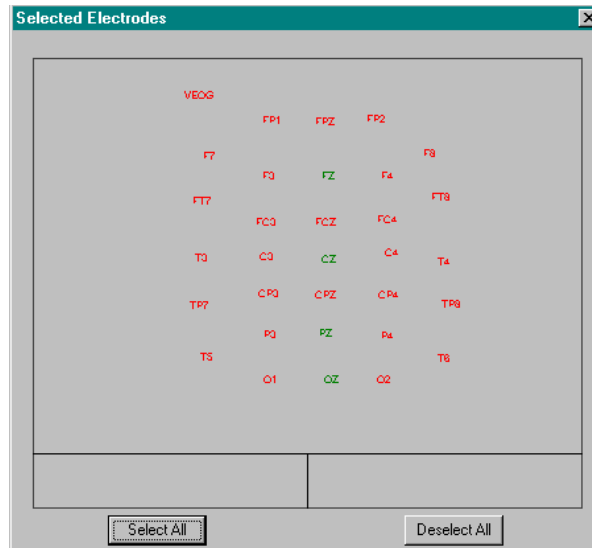


**CHANLIST** - Channel List.  This command will display the standard montage screen, from which you may view or modify the selected electrodes.

**Return.** The selected channels can be returned.

- |   |        |                             |
|---|--------|-----------------------------|
| 1 | list   | List of selected electrodes |
| 2 | string | Title                       |

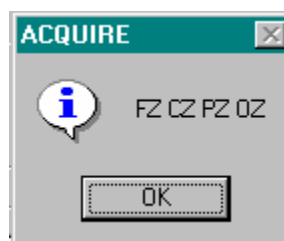
*Example.* CHANLIST {FZ CZ PZ OZ} "Selected Electrodes". You will see a display such as the following.




The "Title" you entered is displayed at the top. You may view the returned contents by using the INSTRUCT command, as follows.

```
set electlist [CHANLIST {fz cz pz oz} "Selected Electrodes"]
INSTRUCT $electlist
```


The contents of the list will be displayed:



**Data Scaling Commands.** These commands are used to scale the data displays automatically or manually.

**AUTOSCALE** - Autoscale.  The AUTOSCALE command will automatically search for the minimum and maximum values in the electrode array. The display min and max values will be set to the found minimum and maximum values, respectively. Electrodes set to be skipped will not be included in the search.


*Example.* After retrieving a data file that you wish to autoscale, place the AUTOSCALE command on a line by itself to scale the data automatically.

**SCALE** - Scale the Displayed Data.  The SCALE command sets the min/max voltage scale range for screen display (EEG and AVG files only), and will not affect the data. This command is useful in setting a series of files to the same display scale range.

1	int	Minimum voltage
2	int	Maximum voltage

*Example.* OPENFILE "c:\\ScanData\\Batch Examples\\MYDATA.avg"  
SCALE -10 10  
SAVEAS "c:\\ScanData\\Batch Examples\\MYDATA.avg"

In this example, the file MYDATA.avg is recalled, scaled to  $\pm 10$  uVs, and overwritten to disk.

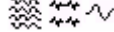
**SCALEX** - Scale the X-axis.  The SCALEX command sets the min/max time (for time domain files) or frequency (for frequency domain files) scale range for the screen display (EEG and AVG files only). If you save the modified file, the changes will be permanent; you cannot go back to the original start/stop times.

1	int	Minimum time point
2	int	Maximum time point

*Example.* OPENFILE "c:\\ScanData\\Batch Examples\\MYDATA.avg"  
SCALEX 0 400  
SAVEAS "c:\\ScanData\\Batch Examples\\NEWDATA.avg"

In this example, the file MYDATA.avg is recalled, the x-axis is rescaled from 0 to 400 ms, and the modified file is saved to a new file.

**File Management Utility Commands.** These are some basic commands that are used throughout the online and offline programs.




**DELETE** - Delete File(s).  This command can be used to delete a file(s) without exiting the program. The filename should be specified with an extension and a full path.

1...n	list	File name(s) ("" will fail)
-------	------	-----------------------------

*Example.* DELETE {{c:\\ScanData\\Batch Examples\\badfile1.cnt}  
{c:\\ScanData\\Batch Examples\\badfile2.avg}}. *Note: if you are deleting a single file, you still need to use double braces - DELETE*

{{<file path and name>}}. If you are using variable substitutions, recall that you cannot use substitutions within braces. In that case, use a format similar to the following:




```
DELETE [list $path\file1.eeg $path\file2.eeg $path\file3.eeg]
```

**NEWNAME** - New File Name.    This command may be inserted at any point in the BATCH file, and is used to rename *a file*.

1	string	File name ("" will fail)
2	string	New file name

*Example.*

```
NEWNAME "c:\\ScanData\\oldname.eeg" "c:\\ScanData\\newname.eeg".
```

**RENAME** - Rename Electrode.    This option is used to rename *electrode labels*. Retrieve the data file, and enter the old name and the new name for the label to be renamed. Save the file with the new labels, if desired.

1	string	Electrode name
2	string	New name (maximum 19 characters)

*Example.* RENAME "Czed" "CZ".

**GET/SET Commands.** The various GET and SET commands offer a wide range of options for returning or modifying information or settings in the data files.

**GETBASELINECOR / SETBASELINECOR** - Get/Set Baseline

Display Correction. Use this command to query the baseline correction (display only) status of the current working file.

**Return.** The GETBASELINECOR command will return either a 0 (no correction applied), or a 1 (baseline correction applied). The SETBASELINECOR command is used to apply or remove the correction.

*Example.* GETBASELINECOR. This command used no parameters. The command will return either a 1 or a 0, meaning the correction has been applied or not. To see the returned value, use a script such as the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\viscpt.cnt"
set basecor [GETBASELINECOR]
INSTRUCT $basecor
```


The returned information will appear as follows:



## SETBASELINECOR

1                      Boolean                      Enable

*Example.* SETBASELINECOR Y. The display baseline correction will be applied.

**GETCHANATTRIBUTE** - Get Channel Attribute.  This command is used to return the channel attributes for a specified channel (all file types).

**Return.** The command will return a 1 if the attribute is set, or a 0 if it is not.


1            string                      Channel label  
2            defined value            Attribute Type (-Artifact, -Fsp, -Hide, -Skip, -Bad, -AutoAdd, -AutoAddLast)

*Example.* GETCHANATTRIBUTE "CZ" -Bad. The program will return a 1 or a 0, depending on whether CZ is set as a Bad channel or not. The following script shows how to view the returned value.

```
OPENFILE "c:\\ScanData\\Batch Examples\\p300.eeg"
set attrib [GETCHANATTRIBUTE CZ -Bad]
INSTRUCT $attrib
```

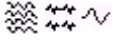
In this example the CZ electrode was not set as a "Bad" channel, so the returned value is 0.



**SETCHANATTRIBUTE** - Set Channel Attribute.  This command is used to set the selected channel attribute, as desired (use with any file type). The command can be used with single, several, or all channels in the file. When using the command online in ACQUIRE, you must place the command after GETAST and before STARTACQUISITION.

1	list	Channel label(s) or "All"
2	defined value	Attribute Type (-Artifact, -Fsp, -Hide, -Skip, -Bad, -AutoAdd, -AutoAddLast)
3	Boolean	Set attribute

*Example.* SETCHANATTRIBUTE {Fp1 Fp2 Fz F7 F8} -Art Y.  
The designated channels will be set as Artifact Rejection channels.

**GETCHANLABEL** - Get Channel Label.  This command is used to retrieve a selected channel label. See also GETCHANNELINDEX.

**Return.** The command will return the channel label.

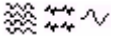
1	string	Electrode number (zero-base index)
---	--------	------------------------------------

*Example.* GETCHANLABEL "12". The command will return the label for channel 12. Note that this is a zero-based index, meaning that the first electrode is 0, not 1. The T3 label that is returned is the 13th electrode in the setup file. To see the returned string, use a script similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set label [GETCHANLABEL "12"]
INSTRUCT $label
```

You will see the label for channel 13.

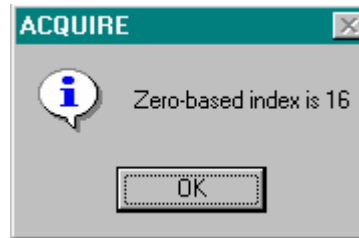


**GETCHANNELINDEX** - Get Channel Index.  This command will return a zero-based index for the selected channel. See also GETCHANLABEL.


1	string	Electrode label.
---	--------	------------------

*Example.* GETCHANNELINDEX "CZ". The zero-based channel index for CZ is returned.

```
OPENFILE "c:\\ScanData\\Batch Examples\\vcp.avg"
set index [GETCHANNELINDEX "CZ"]
INSTRUCT "Zero-based index is $index"
```



The returned value is displayed:

**GETDISPLAYINVERTED / SETDISPLAYINVERTED** - Get/Set Display Inversion.  These commands are used to determine the current status of the Invert Polarity command (GETDISPLAYINVERTED), or to set the display polarity (SETDISPLAYINVERTED).

**Return.** The command will return either a 0 or a 1, where 0 is the polarity setting when the file was recorded, and 1 means the polarity has been inverted.

*Example.* GETDISPLAYINVERTED. This command uses no parameters. A simple script would be as follows:

```
OPENFILE "c:\\ScanData\\Batch Examples\\vcp.avg"
set invert [GETDISPLAYINVERTED]
INSTRUCT $invert
```

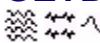
The returned value is displayed as follows:



### SETDISPLAYINVERTED

1                      Boolean              Invert

*Example.* SETDISPLAYINVERTED Y. The polarity will be inverted.

**GETDISPLAYPAGE / SETDISPLAYPAGE** - Get/Set Display Page.  These commands are used to return the number of the display page, or to display a specified display page.

**Return.** The current display page is returned.

*Example.* GETDISPLAYPAGE. This command has no parameters. A simple example for seeing the returned value is as follows (after retrieving a file and displaying a display page):

```
set page [GETDISPLAYPAGE]
INSTRUCT $page
```



## SETDISPLAYPAGE

1                      int                      Display page (zero-based)

*Example.* SETDISPLAYPAGE 1. In this example, the second display page will be shown (zero-based index).

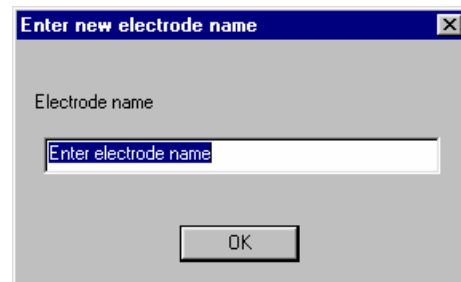
**GETINPUT** - Get Input Text. This command can be used to create a window for entering text that is then used in the BATCH program. One application might be to display a customized Open File window for entering new electrode labels, as shown in the example below.

1	string	Title
2(optional)	string	Prompt
3(optional)	string	Edit default

*Example.* GETINPUT "Enter new electrode name" "Electrode name" "Enter electrode name". The first argument is the title of the window that is displayed. The second argument (optional) is the prompt that appears above the input text field. The third argument (optional) appears in the text field, already highlighted. The example below demonstrates how this may be used.

```
OPENFILE "c:\\ScanData\\Batch Examples\\vep.avg"
set newlabel [GETINPUT "Enter new electrode name" "Electrode
name" "Enter electrode name"]
# Enter Czcd for the new label
RENAME "CZ" "$newlabel"
SAVEAS "c:\\ScanData\\Batch Examples\\newvep.avg"
```

You will see the following display. Enter the new electrode name, such as Czcd, and then the new file will be saved. The change will be made to the working file as well (the retrieved vep.avg file), but it will not



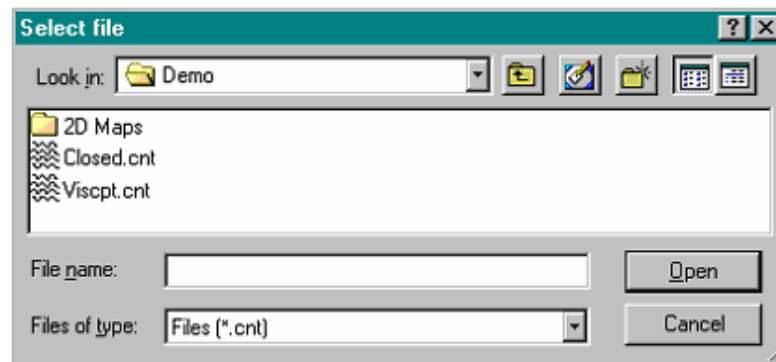





be applied unless you save that file with the modification.

**GETINPUTFILE** - Get Input File. This command is used to display the standard Open File utility display, with optional customization. See also GETOUTPUTFILE.

1	string	String title
2(optional)	string	Extension
3(optional)	string	Initial Path

*Example.* GETINPUTFILE "Select file" ".cnt" "c:\\ScanData\\Batch Examples\\". The Select file screen seen below is opened. The first argument (required) is the title of the display. The second argument (optional) will set the extension in the "Files of type" field. The third argument (optional) defines the default path - the folder that is displayed when the window opens. You may then select the file and proceed with the rest of the BATCH commands.



**GETNUMCHANS** - Get Number of Channels.    This command will return the number of channels in any type of data file.

**Return.** The number of channels is returned.

*Example.* GETNUMCHANS appears on a line by itself, and uses no parameters. The returned value is the number of channels. To see the returned value, use a script similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vep.avg"
set number [GETNUMCHANS]
INSTRUCT $number
```

The number of channels will be displayed.



**GETNUMDISPLAYPAGES** - Get Number of Display Pages. 🌀⚡📶 This command will return the number of Display Pages in any type of data file.

**Return.** The number of display pages is returned.

*Example.* GETNUMDISPLAYPAGES. This command uses no parameters. It will return the number of display pages there are for the data file that has the "focus". To see the returned value, use a script similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\closed.cnt"
set pages [GETNUMDISPLAYPAGES]
INSTRUCT $pages
```

The number of display pages will be displayed.



**GETNUMPOINTS** - Get Number of Points. 🌀⚡📶 This command will return the number of points in a sweep or file (CNT, AVG or EEG files).


**Return.** The number of points is returned.

*Example.* GETNUMPOINTS appears on a line by itself, and uses no parameters. To see the returned value, use a script similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set points [GETNUMPOINTS]
INSTRUCT $points
```

The number of data points will be displayed.



**GETNUMSWEEPS** - Get the Number of Accepted/Rejected Sweeps.  This command will return the number of accepted or rejected sweeps in an AVG or EEG file.

**Return.** The number of accepted and/or rejected sweeps is returned.


1	defined value	Parameter (-Accepted, -Rejected)
---	---------------	----------------------------------

*Example.* GETNUMSWEEPS -Acc. After retrieving an AVG or EEG file, this command will return the number of accepted sweeps. A more complete example is:

```
OPENFILE "c:\\ScanData\\Batch Examples\\p300.eeg"
set acc [GETNUMSWEEPS -Accept]
set rej [GETNUMSWEEPS -Reject]

set report [format "accepted: %d rejected: %d" $acc $rej]
INSTRUCT $report
```

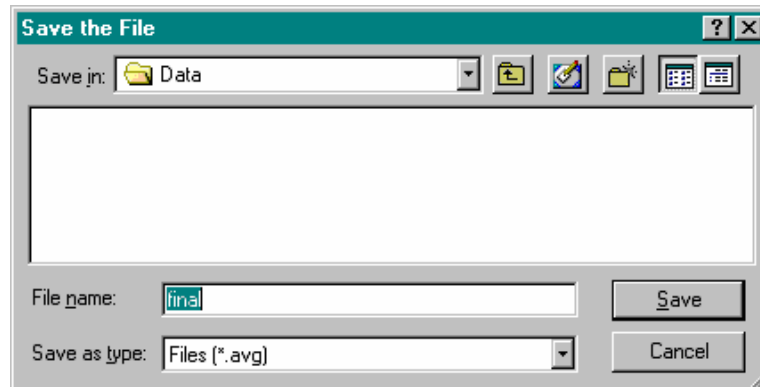
The following information is returned (depending on how you set the threshold in Artifact Rejection).

**GETOUTPUTFILE** - Get Output File.  This commands lets you create and display the standard Save As utility window. See also GETINPUTFILE.


1	string	String title
2(optional)	string	Default extension
3(optional)	string	Default filename
4(optional)	string	Initial path

*Example.* GETOUTPUTFILE "Save the File" "avg" "final" "c:\\ScanData\\Batch Examples". The command has up to 4 arguments. The first (required) sets the title for the window. The second (optional) sets the default file extension (seen in the "Save as type" field). The third argument enters a default file name. The fourth argument

(optional) sets the default path. Click Save or make any changes, then proceed with the rest of the BATCH file.



## **GETSWDISPLAYSPEED / SETSWDISPLAYSPEED** - Get/Set

Display Speed.  These commands are used to return the current number of seconds in the display (the display speed), or set the number of seconds in the display (CNT files only).

**Return.** The number of seconds displayed is returned.

**GETSWDISPLAYSPEED.** This command will return the current number of seconds. To see the returned value, use a script similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\closed.cnt"
set speed [GETSWDISPLAYSPEED]
INSTRUCT $speed
```


The display speed (number of seconds per display screen) will be displayed.



### **SETSWDISPLAYSPEED**

1                      int                      Seconds in display (2-20 are valid entries)

*Example.* SETSWDISPLAYSPEED 10. Ten seconds per screen will be displayed.

**GETSWEEPMAX** - Get Sweep Stop Latency.  This command will return the stop latency of the sweep (in the display that has the focus).

**Return.** The sweep maximum is returned.

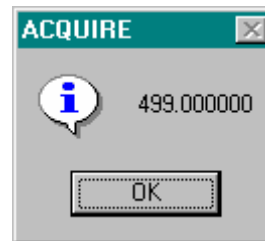
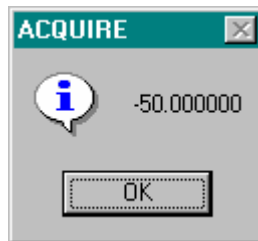
*Example.* GETSWEEPMAX. This command is used on a line by itself, with no parameters. See GETSWEEPMIN for an example.


**GETSWEEPMIN** - Get Sweep Start Latency.  This command will return the start latency of the sweep (in the display that has the focus).

**Return.** The sweep minimum is returned.

*Example.* GETSWEEPMIN. This command is used on a line by itself, with no parameters. The following script will return the MIN and MAX values (sequentially) for viewing:

```
set path "c:\\ScanData\\Batch Examples"  
OPENFILE "$path\\vcp.avg"  
set min [GETSWEEPMIN]  
set max [GETSWEEPMAX]  
INSTRUCT $min ;INSTRUCT $max  
CLOSEALL
```



**HERTZTOPOINT** - Point Number for Hz.  This command is used to return the nearest point index (zero-based) to the Hz value you select (see the example below). "Point" in this case the frequency bin number. The command is used with frequency domain AVG and EEG files only. See also POINTTOHERTZ.

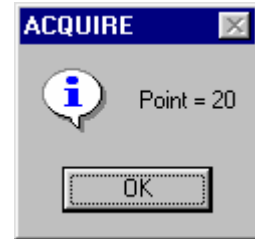
**Return.** The zero-based, frequency bin number nearest the selected Hz value is returned.

1	double	Hz value
---	--------	----------

*Example.* The sequence below retrieves a frequency domain AVG file, and returns the zero-based frequency bin number nearest 10Hz.

```
OPENFILE "c:\\ScanData\\demo\\EEGs\\closedfft.avg"  
set point [HERTZTOPOINT 10]  
INSTRUCT "Point = $point"
```

In this example, "20" is returned.

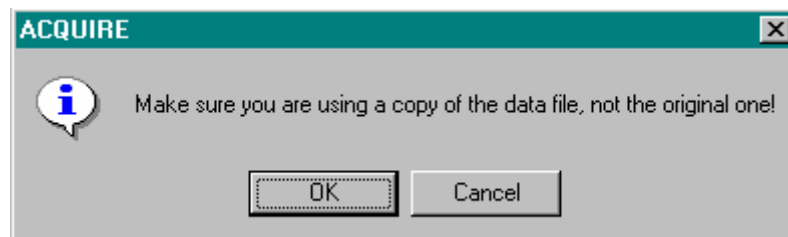


**INSTRUCT** - User Instruction. The INSTRUCT command allows the user to insert messages to be displayed on the screen while the program is in the command mode. These can be used as a prompt to the user. The command may also be used to return requested information.

**Return.** The text string that you enter will be returned (as well as the buttons you specify).

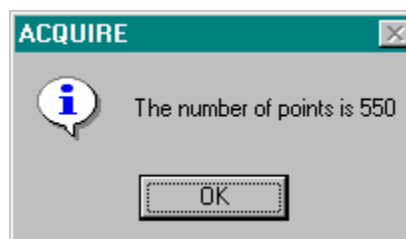
1	string	Instruction
2(optional default =OK)	defined value	Type (OK, OKCANCEL, RETRYCANCEL, YESNO, YESNOCANCEL)


*Example.* INSTRUCT "Make sure you are using a copy of the data file, not the original one!" OKCANCEL. The message will be displayed, with the OK and Cancel options available to the user.



*Example.* To display returned information, create a sequence such as the following:

```
OPENFILE "c:\\ScanData\\Batch Examples\\vcp.avg"
set points [GETNUMPOINTS]
INSTRUCT "The number of points is $points"
```



The second parameter was omitted, resulting in the display of the  button only. You can make the command simpler by using **INSTRUCT \$points**, and the display will contain only the returned value (550).


**ISOFFLINE / ISONLINE** - Online or Offline. These commands are used primarily in conjunction with the **autorun** feature. You may want to have a different set of functions loaded depending on whether you are online (in ACQUIRE) or offline (in EDIT). The commands will return either a 0 or 1, depending on which program you are in (0 means no, and 1 means yes). No parameters are used with either command.

**Return.** Either a 0 or a 1 is returned.

*Example.* See the following example:

```
set online [ISONLINE]
INSTRUCT "The status is $online"
set offline [ISOFFLINE]
INSTRUCT "The status is $offline"
```

If that script is executed from the ACQUIRE program, "online" will be 1, and "offline" will be 0. If executed from the EDIT program, "online" will be 0, and "offline" will be 1.

**LATENCYTOPOINT** - Return Data Point Index for a Given Latency.  This command is used to return the index of a data point at a given latency (AVG and EEG files only). For example, the **GETPOINTDATA** command is used to return the value of a given data point, or set of points. It requires the index number of the data points (not the millisecond values). Knowing the AD rate and the start time of the sweep, you can calculate the index numbers, or, you can use **LATENCYTOPOINT**. Enter the latency of a data point, and the command will return the index number of the point. See also **POINTTOLATENCY**.

**Return.** The number (index) nearest the latency is returned.

1	double	Latency
---	--------	---------

*Example.* **LATENCYTOPOINT 100.** The program will return the number (index) of the data point nearest to the 100th millisecond. The script might look similar to the following (to see the returned value):

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vep.avg"
set point [LATENCYTOPOINT 100]
INSTRUCT $point
```



**MAXIMIZE** - Maximize Window. This command performs the standard Windows Maximize function. Applying it will maximize the display of the window that has the focus. Use **SELECTFILE** to change the focus to the desired window, as needed.

*Example.* **MAXIMIZE**. No additional parameters are used.

**MINIMIZE** - Minimize Window. This command performs the standard Windows Minimize function. Applying it will minimize the display of the window that has the focus. Use **SELECTFILE** to change the focus to the desired window, as needed.

*Example.* **MINIMIZE**. No additional parameters are used.

**PAUSE** - Pause. The **PAUSE** command will halt execution of a BATCH file until the button is pressed. This allows you to perform a variety of interactive functions at the point where the BATCH file is suspended. It can be used with the optional "time to wait". This is useful to delay the execution of the next command for a specified period of time.

1 (optional)                      int                      Time to wait (in ms's)

*Example.* **PAUSE 1000**. This will suspend the BATCH file for 1000ms.  
Please see the acquisition examples below for applications of this command.

**"PICKOFF" commands.** The various "pickoff" commands are used to select segments from a string and create individual variables for each of them. These can then be combined with the Tcl command: format. See the example below (and a Tcl text for more details).

**PICKOFFDIRECTORY** - Pick Off the Path and Folder. The complete path and destination folder will be extracted from the string (includes all backslashes: c:\ScanData\Batch Examples\).

**PICKOFFEXTENSION** - Pick Off the File Extension. The file extension, including the leading "dot" (as in .cnt), will be extracted.

**PICKOFFFILENAME** - Pick Off the Complete File Name. The complete file name, including the extension (as in closed.cnt) will be extracted.



**PICKOFFNAMEONLY** - Pick Off the File Name. The file name, without the extension (as in viscpt) will be extracted.

The single parameter for each is:


1	string	File name and path
---	--------	--------------------

Use is demonstrated in the following script:

```
set path "c:\\scan4.3\\Batch Examples\\viscpt.cnt"
OPENFILE $path
set Filetag [GETINPUT "File Tag" "Enter a tag for the data file, such as:
corrected"]

set Directory [PICKOFFDIRECTORY $path]
set Filename [PICKOFFNAMEONLY $path]
set Ext [PICKOFFEXTENSION $path]

set OutFile [format "%s%s_%s%s" $Directory $Filename $Filetag
$Ext]
SAVEAS $OutFile
OPENFILE $OutFile
```

**POINTTOHERTZ** - Hz for Point Number.  This command is used to return the nearest Hz frequency to the zero-based bin number value you select (see the example below). "Point" in this case the frequency bin number. The command is used with frequency domain AVG and EEG files only. See also HERTZTOPOINT.

**Return.** The Hz value nearest the selected zero-based, frequency bin number is returned.


1	double	Point (frequency bin) value
---	--------	-----------------------------

*Example.* The sequence below retrieves a frequency domain AVG file, and returns the Hz value nearest the selected zero-based frequency bin number.

```
OPENFILE "c:\\scan4.3\\demo\\EEGs\\closedfft.avg"
set Hz [POINTTOHERTZ 20]
INSTRUCT "Hz = $Hz"
```

In this example, "9.765630" is returned.



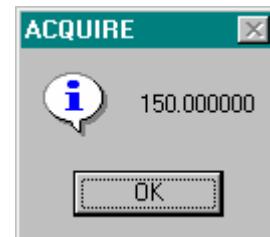
**POINTOLATENCY** - Point Index to Latency.  This command is the complement to the LATENCYTOPOINT command, and is used to return the latency nearest to the data point you specify (EEG and AVG files only). Enter the zero-based index of the data point, and the command will return the latency (in milliseconds) of the nearest data point.


**Return.** The latency nearest the entered data point is returned.

1                      int                      Index of a data point (zero-based index)


*Example.* POINTOLATENCY 200. The program will return the latency of the 200th data point. The script will appear similar to the following:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set latency [POINTTOLATENCY 200]
INSTRUCT $latency
```



**PRINTWAVEBOARD** - Print Waveboard Display.  This command is used to Print the display in the Waveboard (no additional parameters are used).

*Example.* PRINTWAVEBOARD. The Waveboard display is printed.

**READPOS/SAVEPOS** - Read/Save Electrode Position Files.  The READPOS command will read an ASC file, and position the electrodes on the screen according to the new coordinates. The SAVEPOS command will save an ASC file specified by the filename argument that contains the positions of the current electrode displays.

1                      string                      Position file name (or ""; ASC extension)

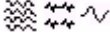
*Example.* READPOS "c:\\ScanData\\Batch Examples\\position.asc". With an EEG or AVG file already in working memory, the position file is used to reposition the electrode channel displays. The file must have the same number of channels and labels as the file in working memory. The position files are created

from the Channel Layout screen. You may wish to use the SAVEAS command afterwards to save the data file with the new positions.


**REM** - Remark. The REM command is used to place text in the BATCH file that is not recognized by the program. It is used interchangeably with the # sign.

REM This is some example text.  
REM After the REM command appears in a line,  
REM any text can follow.


The REM and # commands can be used, for example, to explain a series of commands.

**RESTORE** - Restore Window.  This command performs the standard Windows Restore function. Applying it will restore the display of the window that has the focus to its previous size. Use SELECTFILE to change the focus to the desired window, as needed.

*Example.* RESTORE. No additional parameters are used.

**REVIEW** - Review Sweeps. The REVIEW command has the same function as the PAUSE command - it suspends the BATCH file at the point where the command is inserted. In EDIT, this allows you to, for example, review and edit epoched files for artifact in the interactive single-sweep mode. When used with the "Time to wait" argument, the selected file(s) will be displayed for the designated time span. To terminate the REVIEW command, press the  button on the Batch dialog display. All menu items and Toolbar icons in EDIT are active while the REVIEW command is in effect.

1 (optional)    int                      Time to wait (in milliseconds)

*Example.* REVIEW 500. If no delay is specified, the display will pause until you press the  button (allowing you to, for example, step through the file manually and set each sweep as "accept" or "reject"). The "time to wait" argument is useful in cases where, for example, you want to step automatically through epochs in an EEG file, as shown in the example below.

```
OPENFILE {c:\ScanData\Batch Examples\test.eeg}  
set epochcount [GETEPOCHCOUNT]  
for {set index 0} {$index < $epochcount} {incr index} {  
  SETCURRENTEPOCHINDEX $index  
  REVIEW 1000  
}
```

**SETSPECTDISPLAYSTYLE** - Set Spectral Display Style.  This command is used to select the style of the spectral display - line or histogram. It is used with frequency domain AVG and EEG files.

1                      defined value                      Style (LINE, HISTOGRAM)

*Example.* SETSPECTDISPLAYSTYLE HIST. The file will be displayed as a histogram.

**Subject Information Commands.** These commands are relevant for accessing, modifying and saving subject information.

**READSUB** - Read Subject Information File. The READSUB command will load a subject information file. Subject information files have a .SUB extension, and may be saved manually from ACQUIRE, or with the SAVESUB command.

1                      string                      Subject file name (or "")

*Example.* READSUB "c:\\ScanData\\Batch  
Examples\\subjectfile.sub".

**SAVESUB** - Save Subject Information File. The SAVESUB command will save a subject information screen to disk. Subject information files have a .SUB extension and can be read manually from ACQUIRE or EDIT, or in BATCH with the READSUB command.

1                      string                      Subject file name (or "")

*Example.* SAVESUB "c:\\ScanData\\Batch  
Examples\\subjectfile.sub".

**SUBJECT** - Subject Information Display. The SUBJECT command will display the subject information screen, in which you may enter subject information. This command can be useful in automating the acquisition process. Save the subject information form from the Subject display, or with the SAVESUB command.

*Example.* The SUBJECT command exists on a line by itself, with no additional parameters.

**Substitutions.** One of the more powerful aspects of BATCH files is that you can create a set of operations to be applied to one file, and then apply them to a series of like data files - all in the same BATCH program. This is accomplished by using some form of file name substitution. The INSERT command, as used in the DOS BATCH files,

provides one means. Other, more powerful options are described above in the Variable Substitutions section.

**INSERT** - Insert. The INSERT command allows you to add characters to a file name. When this command appears in the BATCH file, a prompt will appear on the screen requesting you to enter a character string. To insert the string use the % sign in the desired filename. This command can be used to automate a BATCH file in which the same file name will be used in several places throughout the file. Here is an example. Let's say you have several files named XXXfile, where XXX is a number. Select a specific file by entering 123 in the INSERT display, for example, to process the 123file.eeg. This command is included to provide backward compatibility with the previous DOS versions of BATCH. New commands such as GETINPUT, GETINPUTFILE, GETOUTPUTFILE or other substitution methods should be used instead of INSERT when creating new BATCH files.

```
INSERT           ;# Insert the desired text string
OPENFILE "c:\\ScanData\\Batch Examples\\%file.eeg"
BASECOR PRE 0 0 N N "c:\\Scan4.3\\Batch
Examples\\%file.eeg"
```

In this example, the INSERT command will request a character string to be entered from the keyboard by you. If you enter, for example 123, then 123 will be added to the beginning of the file name in the OPENFILE and BASECOR commands in place of the % sign. (This assumes you have a file named 123file.eeg in the indicated folder).

**WRITELOG** - Write Log File. You may create a separate log file containing any text string(s) that you specify in the BATCH file.. The WRITELOG command is used to create the file, and to specify the text string to be written to it. The text string can be any information you want to save in a separate LOG file. If you retrieve the same LOG file again, any new information will be added to the end of the file. (The file is closed automatically).

1	string	File name (or "")
2	string	Text to be included in the LOG file

*Example.* WRITELOG "c:\\ScanData\\Batch Examples\\mylog.log" "Any text entered here will appear in the LOG file". The following script displays the use of the command.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\Batch Examplesfile.eeg"
AVERAGE T N N "" 0 0 0 "" "$path\\average_file.avg"
WRITELOG "$path\\mylog.log" "Any text entered here will appear in the
LOG file"
```

WRITELOG "\$path\\mylog.log" "Text written to the same file will be added at the end of the file"  
CLOSEFILE

**ZOOM Commands.** These commands are used to control the zoom operations in ACQUIRE or EDIT.

**TOGGLEZOOM** - Toggle Zoom. This command is used to toggle between zoomed in and zoomed out electrode displays.

1                      string                      Electrode name

*Example.* TOGGLEZOOM FZ. If the selected electrode is not already displayed full-size, this command will make it full-size. Repeating the command will minimize the display.

**ZOOMIN** - Zoom In. The ZOOMIN command will zoom-in and display a waveform/spectrum at the site named in the electrode argument. This command is used with epoched or averaged data files.

1                      string                      Electrode name

*Example.* ZOOMIN "CZ". After retrieving an epoched or averaged data file, the CZ electrode display is zoomed to a full-size display.

**ZOOMOUT** - Zoom Out. The ZOOMOUT command will zoom-out and display a waveform/spectrum at the site named in the electrode argument. This command is used with epoched or averaged data files.

1                      string                      Electrode name

*Example.* ZOOMOUT "CZ". After retrieving an epoched or averaged data file, the full-sized CZ electrode display is minimized.

**Discontinued DOS Commands for ACQUIRE and EDIT.** These DOS commands were used with the DOS programs, and are now obsolete and no longer recognized.

ACQUIRE  
DRAW  
EDIT  
GETCLIP  
MAPGEN  
QUIT  
RENAMECLIP  
SAVESCREEN  
STATS

SYSTEM  
TAGOFFCLIP  
TAGONCLIP  
WINDOW

## Tcl Commands for ACQUIRE

---

**CALIB** - Calibration. The CALIB command will perform an automatic calibration procedure and display the calibration values. You have the option to accept or reject them.

*Example.* The CALIB command exists on a line by itself, with no additional parameters.

**CLEARAVG** - Clear Average. ~ This command is used to clear the average in an online average, and to restart the average with the next accepted sweep. The command is not applicable when you have multiple online sorted averages - just the single average you see when recording in Average Acquisition mode.


*Example.* CLEARAVG. The CLEARAVG command exists on a line by itself, with no additional parameters.

**\*CONFIGUREBLINKNOISEREDUCTION** - Configure Gradient/Blink Noise Reduction Routine. ☼ This is one of the commands used with the artifact reduction add-on software (to access it you must be under warranty or a maintenance contract; contact sales@neuro.com or techsup@neuro.com for details). The CONFIGUREBLINKNOISEREDUCTION command is used to remove VEOG blink artifact during the acquisition of CNT files. See also ENABLEBLINKNOISEREDUCTION. The same reduction may be applied offline; see BLINKNOISEREDUCTION.

1	double	Epoch start time (ms)
2	double	Epoch stop time (ms)
3	integer	Number of averages
4	Boolean	Use external trigger
5	double	Voltage threshold (uV; ignored if #4 is true)
6	defined value	Trigger direction (Positive, Negative; ignored if #4 is true)
7	string	Artifact channel label (ignored if #4 is true)
8	int	External trigger code (ignored if #4 is false)
9	Boolean	Enable artifact rejection
10	double	Artifact min (uV)
11	double	Artifact max (uV)
12	Boolean	Enable bipolar reference


13	string	Bipolar reference channel label
14	Boolean	Enable high pass filter
15	double	High pass filter value

*Example.* CONFIGUREBLINKNOISEREDUCTION -50 300 10 N 200 P "VEOG" X N 0 0 N "" N x. In the example, intervals from -50 to 300 ms are used, with the trigger occurring when there is a positive deflection of 200 uVs from the VEOG channel. No Artifact Rejection, bipolar reference, or high pass filter are used.

**\*CONFIGUREEKGNOISEREDUCTION** - Configure EKG Noise Reduction Routine.  This is one of the commands used with the artifact reduction add-on software (to access it you must be under warranty or a maintenance contract; contact sales@neuro.com or techsup@neuro.com for details). The CONFIGURE-EKGNOISEREDUCTION command is used to remove ballistocardiogram or other EKG artifact during the acquisition of CNT files. See also ENABLEEKGNOISEREDUCTION. The same reduction may be applied offline; see EKGNOISEREDUCTION.

1	double	Epoch start time (ms)
2	double	Epoch stop time (ms)
3	integer	Number of averages
4	Boolean	Use external trigger
5	double	Voltage threshold (uV; ignored if #4 is true)
6	defined value	Trigger direction (Positive, Negative; ignored if #4 is true)
7	string	Artifact channel label (ignored if #4 is true)
8	int	External trigger code (ignored if #4 is false)
9	Boolean	Enable artifact rejection
10	double	Artifact min (uV)
11	double	Artifact max (uV)
12	Boolean	Correct trigger channel
13	Boolean	Enable bipolar reference
14	string	Bipolar reference channel label
15	Boolean	Enable high pass filter
16	double	High pass filter value

*Example.* CONFIGUREEKGNOISEREDUCTION -50 200 10 N 200 P "EKG" x N 0 0 N N "" N x. In the example, intervals from -50 to 200 ms are used, with the trigger occurring when there is a positive deflection of 200 uVs from the EKG channel. No Artifact Rejection, bipolar reference or high pass filter are used, and the trigger channel is not corrected.

**DODCCORRECT** - Do a DC Correction.  This command is used to perform a planned DC correction (SynAmps only, with a DC high pass filter setting). You might




wish to use this command at the beginning of a recording, or prior to Impedance testing, to remove any DC offset or drifting.

*Example.* DODCCORRECT. The DODCCORRECT command exists on a line by itself, with no additional parameters.

**DOIMPEDANCE** - Go into Impedance Testing. This command will initiate impedance testing. The display must be closed manually in order to continue with the BATCH file. See also HIGHIMPEDANCE below.


*Example.* DOIMPEDANCE. The DOIMPEDANCE command exists on a line by itself, with no additional parameters.


**\*ENABLEBLINKNOISEREDUCTION** - Enable Gradient/Blink Noise Reduction Routine.  This is one of the commands used with the artifact reduction add-on software (to access it you must be under warranty or a maintenance contract; contact sales@neuro.com or techsup@neuro.com for details). The ENABLEBLINKNOISEREDUCTION command is used to initiate the reduction routine. The routine is configured using CONFIGUREBLINKNOISEREDUCTION.

1	Boolean	Enable the routine
---	---------	--------------------

*Example.* ENABLEBLINKNOISEREDUCTION Y. The reduction routine is initiated at the point in the file where this command is placed. The following is how the BATCH sequence might appear.

```
CONFIGUREBLINKNOISEREDUCTION -50 300 10 200 P
"VEOG" N 0 0 Y
STARTACQUISITION
PAUSE
STARTRECORDING ""
PAUSE
ENABLEBLINKNOISEREDUCTION Y
PAUSE
STOPRECORDING
STOPACQUISITION
```

The PAUSEs, as described below, are used to suspend the BATCH program until you click the  button.


**\*ENABLEEKGNOISEREDUCTION** - Enable EKG Noise Reduction Routine.  This is one of the commands used with the artifact reduction add-on software (to access it you must be under warranty or a maintenance contract; contact sales@neuro.com or techsup@neuro.com for details). The ENABLEEKGNOISE-

REDUCTION command is used to initiate the reduction routine. The routine is configured using CONFIGUREEKGNOISEREDUCTION.

1	Boolean	Enable the routine
---	---------	--------------------

Example. ENABLEEKGNOISEREDUCTION Y. The reduction routine is initiated at the point in the file where this command is placed. The following is how the BATCH sequence might appear.

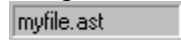
```
CONFIGUREEKGNOISEREDUCTION -50 300 10 200 P
"VEOG" N 0 0 Y Y
STARTACQUISITION
PAUSE
STARTRECORDING ""
PAUSE
ENABLEEKGNOISEREDUCTION Y
PAUSE
STOPRECORDING
STOPACQUISITION
```

The PAUSEs, as described below, are used to suspend the BATCH program until you click the  button.

**GETAST** - Get Setup File. Use this command to load an AST setup file in ACQUIRE. If the file name is omitted, you will be prompted to enter one.

1(optional)	string	AST file name (or "")
-------------	--------	-----------------------

Example. GETAST "c:\\scan4.3\\setup files\\myfile.ast". The setup file will be opened. You can see the current setup file in the far right side of the Status bar




**HIGHIMPEDANCE**- High Impedances Mark Bad or Hide. This command will automatically mark as Bad or Hide any channels that have impedances beyond a threshold you set.

1	int	Impedance limit (kOhms)
2	defined value	Mark as (-Bad, -Hide)
3	Boolean	Reset good electrodes






Example. HIGHIMPEDANCE 25 -Bad N. In this example, any channel with a measured impedance in excess of 25 kOhms will be marked as "Bad" channels. The third parameter will reset previously marked Bad or Hide channels as good channels (removes Bad or Hide flag).

**PAUSEDISPLAY** - Pause Display.  This command is used to pause or un-pause the display without interrupting data storage.


1                      Boolean                      Pause display

*Example.* PAUSEDISPLAY Y. The display will be paused until you click the Resume  button.


A typical sequence used in ACQUIRE might be as follows. The first PAUSE command suspends the BATCH file while you monitor the incoming EEG.

Clicking the  button displays the Save As window, allowing you to enter a file name and path for the output file, then data storage will begin. Clicking the  button starts the BATCH file again. The second PAUSE command then suspends the BATCH file while you monitor the incoming EEG as it is being stored. Clicking  goes on to the next command: PAUSEDISPLAY Y, which pauses the display without interrupting data storage. Clicking  continues the data display (and storage continues). The last PAUSE lets the storage continue, while suspending the BATCH sequence until you click  once again. Then the output file will close and acquisition display will close.


```
STARTACQUISITION
PAUSE
STARTRECORDING
PAUSE
PAUSEDISPLAY Y
PAUSE
PAUSEDISPLAY N
PAUSE
STOPRECORDING
STOPACQUISITION
```

**RESTART** - Restart Acquisition. This command serves the same purpose as the Restart icon  on the ACQUIRE Toolbar, and it is used to restart the acquisition process.

*Example.* RESTART. The command is used on a line by itself with no parameters.

**STARTACQUISITION** - Start Acquisition. Use this command to initiate viewing of the incoming signals. (It has the same functions as the older VIEW command). You may want to place a PAUSE command after it. This will let the incoming signals scroll until you want to begin data storage (then click the  button).

*Example.* STARTACQUISITION. See the sample sequence of commands above and below.

**STARTRECORDING** - Start Recording. Use this command to initiate data storage. If a file name is specified then the data will be stored to the specified file. It is important to use the correct extension for the type of data being stored. If no file name is specified, a Save As screen will appear allowing you to enter a path and file name. You may want to place a PAUSE command after it. This will let data storage continue until you want to stop it (then click the  button).

l(optional)                      string                      File name (or "")

*Example.* The STARTRECORDING "c:\\ScanData\\Batch Examples\\test.cnt". In the example, the data will be written to the designated file. If you leave the parameter out, you will see the Save As display, in which you may select a path and enter a file name.

**STOPACQUISITION** - Stop Acquisition. Use this command to close the acquisition display.



*Example.* The STOPACQUISITION command exists on a line by itself, with no additional parameters. See the sample sequence of commands above and below.

**STOPRECORDING** - Stop Recording. Use this command to suspend data storage.

*Example.* The STOPRECORDING command exists on a line by itself, with no additional parameters. See the sample sequence of commands above and below.

**Various Acquisition Script Examples.** The typical sequence in recording data from a subject is to recall a setup file, enter patient information, open an EEG file and record the data. Here is an example of a BATCH file that would perform these steps automatically:

```
GETAST "c:\\ScanData\\Setup Files\\QuikCap32.ast"
SUBJECT
DOIMPEDANCE
STARTACQUISITION
PAUSE
STARTRECORDING
PAUSE
STOPRECORDING
STOPACQUISITION
```

In this example, the setup file is read by the GETAST command and it configures the system with the desired parameters. Next, the SUBJECT command brings up the subject information screen, into which you may enter the relevant information. The DOIMPEDANCE command displays the impedance screen for impedance testing. The STARTACQUISITION command has the same function as the green arrow in ACQUIRE (to start viewing but not storage). The PAUSE command afterward allows the display to continue until you click the  button. The STARTRECORDING command displays the Save As utility screen to enter the file name and path, then initiates data storage. The PAUSE command then lets the display and the storage continue until you click the  button. The STOPRECORDING command then stops data storage, although the display will continue. The STOPACQUISITION command closes the acquisition display.

If you want to suspend data storage, then manually resume storage to the SAME data file, the script would be similar to the following:

```
STARTACQUISITION
PAUSE
STARTRECORDING
PAUSE
STOPRECORDING
PAUSE
STARTRECORDING
PAUSE
STOPRECORDING
STOPACQUISITION
```

You may also want to resume storage automatically, saving sections for a specified time span:

```
STARTACQUISITION
PAUSE ;# click resume when ready to start saving data, enter file name
STARTRECORDING
PAUSE 5000 ;# records for 5 seconds
STOPRECORDING
PAUSE 10000 ;# displays data without saving for 10 seconds
STARTRECORDING
PAUSE 5000 ;# records for 5 seconds to the same file
STOPRECORDING
PAUSE 10000 ;# displays data without saving for 10 seconds
STARTRECORDING
PAUSE 5000 ;# records for 5 seconds to the same file
STOPRECORDING
STOPACQUISITION
```

To resume storage to a different data file, the script would be similar to the following:

```
STARTACQUISITION
PAUSE
STARTRECORDING
PAUSE
STOPRECORDING
PAUSE
RESTART
STARTRECORDING
PAUSE
STOPRECORDING
STOPACQUISITION
```

Again, the **PAUSE** commands pause the BATCH file, not the operations in **ACQUIRE**. If you have a **PAUSE** line following the **STARTRECORDING** command, the recording continues - only the BATCH file operations are paused.

**VIEW** - View Incoming Signals. Use this command to initiate viewing of the incoming signals. (It has the same functions as the newer **STARTACQUISITION** command which should be used instead). See also **STARTACQUISITION**.

**Discontinued DOS commands for ACQUIRE.** The following commands were used in the DOS BATCH files from SCAN 3 and SCAN 4. They are now either superceded by other commands, or they have been discontinued altogether. Some of the DOS commands were used for OPENing files in preparation for data acquisition. The preferred commands now are in the Data Acquisition Commands section above.

AVGSAVE - Save Average File. This command is superceded by the SAVEAS command.

CLEAR - Clear. The CLEAR command has been discontinued.

CNTSAVE - Save Continuous File. This function has been superceded by the STARTRECORDING command.

EEGSAVE - Save EEG File. This function has been superceded by the STARTRECORDING command.

OPENCNT - Open CNT File. This function has been superceded by the STARTRECORDING command.

OPENEEG - Open EEG File. This function has been superceded by the STARTRECORDING command.

RECORD - Record data. This command has been superceded with the STARTRECORDING command.




SAVEAVG - Save AVG File. This function has been superceded by the SAVEAS command.

## Tcl commands for EDIT

---

The commands for the EDIT, or the offline analysis part of the program, have been grouped in places according to related function. For example, all of the commands related to Peak Detection are grouped together. If you are converting your SCAN 3 or SCAN 4 CMD files, you will find that some commands have been discontinued, others still function, and many have been superceded by newer commands that use the functionality in the latest versions of the SCAN software. We have included in the information below which commands have been discontinued and superceded, as well as what the new commands are. Note that unlike the SCAN 3 BATCH command manual, where ALL of the commands were in alphabetical order by program, the ones below are grouped to an extent by functionality. For example, to find the ZSCORE command, look in Statistical Comparisons. There are many new GET and SET commands, and these are grouped together. Commands such as BASECOR, that have related EXCLUDEFORBASECOR and RESETFORBASECOR commands, have the related commands grouped together. (An alphabetical list of commands for ACQUIRE/EDIT, ACQUIRE, EDIT, and discontinued commands is found at the end of this manual).

As you use the various Transform commands, you will find, just like you see in Point&Click mode, that some transforms require a new output file, while others affect the file in working memory. Whether or not a new output file is required is determined largely by whether or not the entire file is loaded in working memory. AVG files, for the most part, are loaded entirely into memory. When you perform a transform such as BASECOR, it is the file in working memory that is modified. There is no requirement for an output file (although you will have the opportunity to save the modified file as a separate file). Similarly, some commands that affect EEG and CNT files, such as those that modify the event table, do so without requiring an output file (the event table is loaded into working memory). On the other hand, many of the operations on EEG and CNT files require an output file (the data files are too large to fit in working memory). For example, performing BASECOR on an epoched file requires an output file. We will point out whenever possible whether you will need to save the output file or not. You will always have the option to save the modified file when you attempt to close it manually, but it would be better to close them in the BATCH file (see SAVEAS). (Note that CLOSEFILE will not prompt you to save a modified file - you will be prompted when you close the file manually).

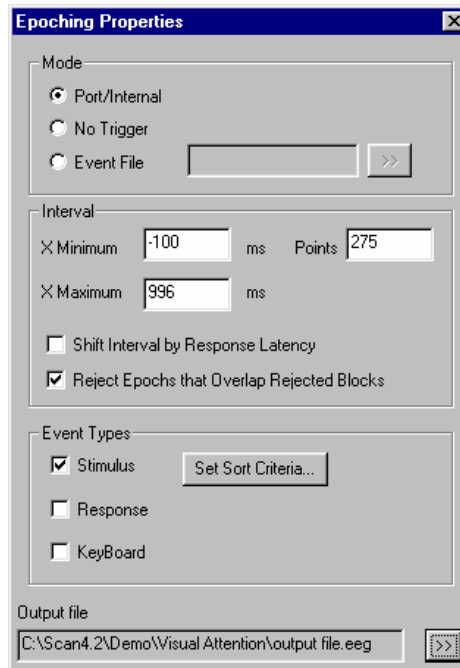
At the end of each Command name, you will see one or more symbols indicating the types of files for which the command may be applied. These are:  for CNT files,  for EEG files, and  for AVG files.

### Auto-write Feature

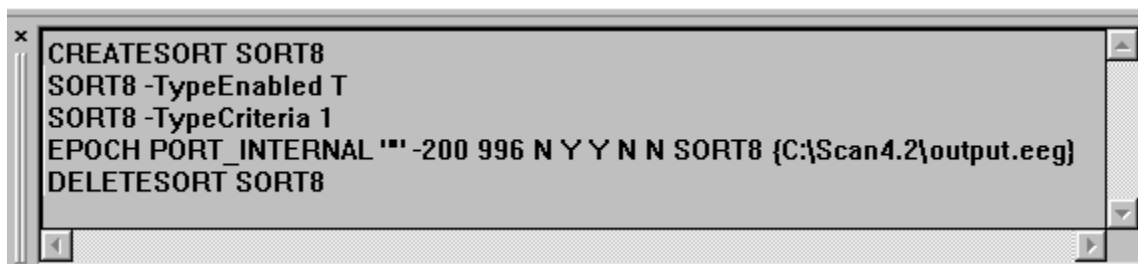
Beginning with SCAN version 4.3, we have added an "auto-write" feature. Whenever you execute a transform in point-and-click (P&C) mode, the Tcl BATCH command will be created automatically, and displayed in the History

toolbar field. From there you can copy the line into Wordpad or Notepad in the batch file you are creating.

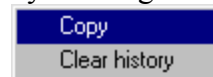
For example, open a CNT file and Epoch it using the transform dialog box in P&C mode.



In this example, we sorted for type codes of 1. The required lines are created automatically, including the "sort" file lines, and displayed in the History toolbar field.



These lines can then be copied and pasted into the BATCH file you are creating by clicking the right mouse in the History region, and selecting the



option.

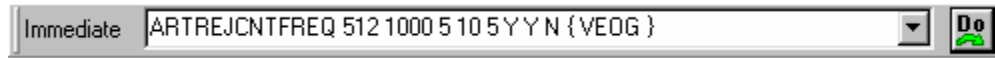
```
CREATESORT SORT8
SORT8 -TypeEnabled T
SORT8 -TypeCriteria 1
EPOCH PORT_INTERNAL "" -200 996 N Y Y N N SORT8
{C:\Scan4.3\output.eeg}
```



## DELETESORT SORT8

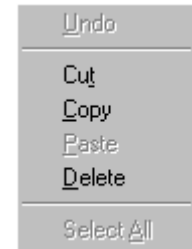
This will save you a lot of time and insure that the correct format is created.

Note that many of the commands are also written to the Immediate command line.





These can be run one line at a time by clicking the Do button. Right click on the Immediate line to see the following option list.


From that list you may select Cut, Copy, Paste, Delete, Select All or Undo (options are active under different conditions). These commands are useful for copying the lines to a BATCH file you are creating, from the BATCH file to the Immediate line, and so on.



## "\_EX" or EXtended Commands

Beginning with SCAN 4.3, there are several "\_EX", or EXtended commands, including BASECOR\_EX, DETREND\_EX, FILTER\_EX, FSP\_EX, and PEAKDETECTION\_EX, and several others. These commands were created because additional functionality was written into the original commands, yet we wished to preserve the original commands for those who already have BATCH files using them. The added parameters in most cases allow you to specify channels to be included *within the command using a list parameter*. With the original commands, you had to use additional commands, such as EXCLUDEFORBASECOR, EXCLUDEFORDETREND, etc., to select channels to be excluded. With the "\_EX" commands, you simply use a list of channels that you want to be *included*. Depending on your specific needs, you may find it easier to use one approach or the other. In other cases, such as COHERENCE\_EX, one or more new sorting parameters have been added. Note that the "auto-write" feature, described immediately above, uses the "\_EX" commands, and it will create the list of included electrodes automatically based

on what you select with the   button, or All.

**ADD** - Add.  The Add command is used to perform a point by point mathematical addition of the selected AVG file to the one specified by the first parameter and create a new output file (opposite of the **SUBTRACT** command). If you start with an EEG file, the selected AVG file will be added to each sweep.

1	string	Filename to add (or "")
2	string	Output file name (or "")

*Example.* ADD "c:\\ScanData\\Batch Examples\\addfile.avg"  
"c:\\ScanData\\newfile.avg". Here are some additional scripts using ADD.

```
#THIS SET OF BATCH COMMANDS ILLUSTRATES SIMPLE ADDITION.
#A VARIABLE CALLED PATH IS DEFINED TO SHORTEN AND SIMPLIFY FILE
#LOCATION ON THE COMPUTER

#SPECIFIES PATH TO DATA FILES
set path "e:\\ScanData\\Batch Examples\\Add and Subtract Transform"
set ExtSum "_Sum"


#FIRST LINE OPENS THE FILE LOCATED IN THE PATH ($PATH) AS DEFINED
#IN THE SET PATH "" COMMAND
OPENFILE "$path\\100HzNeg.avg"

#SECOND LINE DESIGNATES THE FUNCTION (ADD) AND THE FILE TO BE ADDED
ADD "$path\\100HzPos.avg" "$path\\100HzPosNeg_Sum.avg"

#THIRD OPENS THE RESULTS OF THE ADD TRANSFORM
OPENFILE "$path\\100HzPosNeg_Sum.avg"

#IF THE ABOVE FILES ARE INVERTED, THE SUM WAVEFORM FOR THIS EXAMPLE = 0
OPENFILE "$path\\100HzNeg.avg"
ADD "$path\\100HzNeg.avg" "$path\\100HzNegNeg_Sum.avg"
OPENFILE "$path\\100HzNegNeg_Sum.avg"


#IF THE ABOVE FILES ARE IDENTICAL THE SUM WAVEFORM WILL BE
#TWICE THE AMPLITUDE OF EACH ELEMENT IN THE ADDITION
```

**ADDANNOTATION** - Add Annotation.  This command is used to add an annotation to a CNT file at a designated point.

1	string	Annotation text
2	int	Point offset. (zero-based index)


*Example.* ADDANNOTATION "Condition 1" 5000. The comment will be added at the 5000th point in the file.

**Appending Data Files.** This section contains commands used to merge data files together.

**APPEND** - Append.  The APPEND command inserts an additional channel(s) from the file specified by filename to the working file. This command can be used to build a new electrode array from a series of smaller separate recordings. APPEND is used with AVG files.


1	string	Input file (or "")
2	list	Channels to be included, or "All"
3	string	Output file (or "")

*Example.* APPEND {c:\ScanData\secondfile.avg} {P3 P4 C3 C4} {c:\ScanData\finalfile.avg}. Braces MUST be used with this command (not quotes). With braces, you MUST use single slashes in the path (\). Retrieve a data file first, then use the APPEND command to select channels from a second file to add to it, and to create an output file.

**APPENDRECORDING** - Append Recording.  APPENDRECORDING allows you to attach one or more CNT files together into a single CNT file. All files, with their paths, should be listed. Use the OPENFILE command before the APPENDRECORDING command to retrieve the first file, then append the additional file(s) to it.


1	list	File(s) to append ("" will fail)
2	string	Output file name (or "")

*Example.* APPENDRECORDING {{c:\ScanData\file1.cnt} {c:\ScanData\file2.cnt} {c:\ScanData\file3.cnt}} {c:\ScanData\outputfile.cnt}. Braces (not quotes) should be used with commands having the List variable type. With braces, you MUST use single slashes in the path (\). In this example, three CNT files are appended together to form a 4th output file.

**CONCATCNT** - Concatenate CNT files.  This is another way to combine CNT files aside from APPENDRECORDING. The difference is that you do not need to use OPENFILE first - just specify the files to be appended, and the output file name.

1	list	File(s) to append ("" will fail)
2	string	Output file name (or "")

*Example.* CONCATCNT {{c:\ScanData\file1.cnt} {c:\ScanData\file2.cnt} {c:\ScanData\file3.cnt}} {c:\ScanData\outputfile.cnt}. In this example, three CNT files are appended together to form a 4th output file.


**MERGE** - Merge.  This command is used to append one or more EEG files to a previously opened file. The command is applied only to epoched (EEG) files.

1	list	Files to merge ("" will fail)
2	string	Output file name (or "")

*Example.* MERGE {{c:\ScanData\file1.eeg} {c:\ScanData\file2.eeg} {c:\ScanData\fileN.eeg}} {c:\ScanData\outputfile.eeg}. Braces MUST be used with this command (not quotes). With braces, you MUST use single slashes in the path (\). The selected files are merged together to form the single output file.

*Note:* Substitutions CANNOT be used within braces. To use variable substitutions with any LIST parameter, use a format similar to the following (with the braces omitted):

```
MERGE [list $path\file1.eeg $path\file2.eeg $path\file3.eeg]
$path\outputfile.eeg
```

**AREAREPORT** - Area Report.  The AREAREPORT is used to create an ASCII file (DAT extension) that has the Sum, Mean or Area information for the interval you specify. This replaces the **AREA** command from the DOS version.



1	string	Report file name (or "")
2	defined value	Computation type (SUM, MEAN, AREA)
3	Boolean	Rectify
4	Boolean	Use entire interval
5	double	Start latency (ignored if param 4 is YES)
6	double	End latency (ignored if param 4 is YES)

*Example.* AREAREPORT "c:\\ScanData\\Batch Examples\\area report.dat" AREA N N 200 400. In this example, the area (we could have simply entered A) is computed for the interval between 200 and 400 ms. (Notice that the interval boundaries are entered in milliseconds). If you enter a Y for number 4 (to use the entire interval), you still need to enter 0's in the 5th and 6th fields as place holders. The following examples creates the DAT files for two data files.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\p300rare.avg"
AREAREPORT "$path\\area report.dat" AREA N N 200 400
OPENFILE "$path\\p300freq.avg"
AREAREPORT "$path\\area report.dat" AREA N N 200 400
CLOSEALL
```

**Artifact Correction/Rejection.** This section includes both the ocular artifact correction and the automatic artifact sweep rejection commands.



**Artifact Correction** (ocular artifact reduction). The single command used now is ARTCOR; the commands that were used to set various parameters before running the ARTCOR command in the DOS version are no longer needed .

**ARTCOR** - Ocular Artifact Reduction.   The ocular artifact reduction transform is used to subtract out blink or other regularly occurring artifacts (refer to the EDIT manual for details). It is used preferably with CNT files, but may be used with EEG files as well.

1	defined value	Trigger direction (Positive, Negative)
2	double	Threshold percentage
3	int	Minimum sweeps
4	double	Sweep duration
5	string	Blink channel
6	defined value	Output switch (LDR, LDR+CNT)
7	string	LDR file name
8	string	Output CNT file (if param 6 is LDR+CNT)
9	Boolean	Review maxima (CNT files only)
10	Boolean	Review blinks (CNT files only)

*Example.* ARTCOR POS 10 20 200 "VEOG" LDR+CNT  
"\$path\artcor.ldr" "\$path\Correctedfile.cnt" Y Y. In this example, the ocular artifact correction is applied with a positive going threshold at 10%, with 20 minimum sweeps having a duration of 200ms. VEOG is label of the blink channel. The maximum and individual sweeps will be reviewed, and the LDR and corrected CNT files are saved.

**Artifact Rejection.** This transform is used to reject automatically sweeps in which the voltage exceeds a specified threshold. The principle command is ARTREJ.

**ARTREJ** - Artifact Rejection.   The Artifact Rejection option will automatically reject sweeps in which the voltage in a designated channel(s) exceeds defined criteria. The GETCHANATTRIBUTE / SETCHANATTRIBUTE commands can be used to set which channels will be used for artifact rejection.

1	defined value	Operation type (CRITERIA or REJCRITERIA*, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	Boolean	Use entire interval
3	double	Start reject interval (ignored if param 2 is YES)
4	double	Stop reject interval (ignored if param 2 is YES)
5	Boolean	Recompute
6	double	Minimum amplitude
7	double	Maximum amplitude
8	Boolean	Exclude bad channels
9	Boolean	Exclude skipped channels

\* In SCAN 4.3, the "CRITERIA" option was available. In SCAN 4.3, the "REJCRITERIA" term was added to balance the new "ACCCRITERIA" option. In SCAN 4.3 the program will recognize CRITERIA as it did in SCAN 4.2, as well as REJCRITERIA - the program interprets the two equivalently.

*Example.* ARTREJ REJCRIT Y 0 0 N -75 75 Y Y. In this example, sweeps will be rejected based on the voltage criteria. The entire interval is used, the sweeps are not being recomputed, the criteria are set for -75 and 75 uVs, and Bad and Skipped channels are being excluded. Note that 0's are entered as place holders in the 3rd and 4th fields even though they are ignored (because #2 is true).

**ARTREJ\_EX** - Artifact Rejection, extended. \*\*\* The ARTREJ\_EX command is the same as the ARTREJ command, except that it allows you to specify which channels should be the artifact rejection channels and the options to exclude bad or skipped channels are removed (since they are really not needed when you select the channels to include).

1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	Boolean	Use entire interval
3	double	Start reject interval (ignored if param 2 is YES)
4	double	Stop reject interval (ignored if param 2 is YES)
5	Boolean	Modify status
6	double	Minimum amplitude
7	double	Maximum amplitude
8	Boolean	Exclude Bad channels
9	Boolean	Exclude Skipped channels
10	list	List of artifact rejection channels


*Example.* ARTREJ\_EX REJCRITERIA Y x x Y -50 50 Y Y { FP1 F7 T3 T5 FPZ FT7 FP2 F8 T4 T6 FT8 }. In this example, the indicated channels will be monitored for voltages in excess of +/- 50uVs.

**ARTREJCNT** - Artifact Rejection for CNT files (time domain correction). ☼ The routine scans all, or specified channels for voltages exceeding the criteria set in the Min and Max parameters. When a voltage exceeds either criterion, a block of data is rejected according to the limits set in the Block Interval parameters. The interval of the rejected block is defined by the pre-Artifact and Post-Artifact time spans. The Refractory period is a span of time following the artifact during which additional artifacts will not be detected. (Please see the EDIT manual for more details).

1	double	Pre-artifact time span (ms)
2	double	Refractory period (ms)


3	double	Post-artifact time span (ms)
4	Boolean	Recompute
5	double	Min amplitude (uV)
6	double	Max amplitude (uV)
7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skip channels
9	list	List of rejection electrodes

*Example.* ARTREJCNT 100 500 100 Y -100 100 Y Y { ALL }.  
In this example, detection of a voltage in excess of  $\pm 100$ uVs, from any channel, will result in the rejection of the block of data from 100ms before to 100ms after the point of the artifact, with a refractory period of 500ms. Bad and Skipped channels are excluded.

**ARTREJCNTFREQ** - Artifact Rejection for CNT files (frequency domain correction).  This routine performs a series of FFTs, and then rejects epochs when the power in a specified range exceeds a specified voltage threshold. The span of the epochs is determined by the Epoch Points parameter, and that will interact with the AD rate used when the file was recorded.

1	integer	Epoch Points (power of 2)
2	double	Slide interval duration (ms)
3	double	Start frequency
4	double	Stop frequency
5	double	Threshold (uV)
6	Boolean	Recompute
7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skipped channels
9	list	List of electrodes to include

*Example.* ARTREJCNTFREQ 512 100 5 10 5 Y Y Y { VEOG }.  
In this example, FFTs are performed for each sweep containing 512 points. The interval slides every 100ms. Power in the 5-10Hz band, in the VEOG channel, that exceeds 5 uVs will result in a rejected block. Blocks will be recomputed using these criteria; Bad and Skipped channels are excluded.

**ARTREJFREQ** - Artifact Rejection based on frequency band amplitude.   
The ARTREJFREQ command uses the Frequency Domain criteria to accept/reject sweeps.


As described in the EDIT manual, an FFT is computed (whether the numbers of points is a power of 2 or not), and sweeps are then accepted or rejected on the basis of the amplitude



criteria you select within a specified band width. The **GETCHANATTRIBUTE / SETCHANATTRIBUTE** commands can be used to set which channels will be used for artifact rejection.


1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	double	Start reject frequency
3	double	Stop reject frequency
4	Boolean	Recompute
5	double	Maximum amplitude
6	Boolean	Exclude Bad channels
7	Boolean	Exclude Skipped channels

*Example.* **ARTREJFREQ REJCRITERIA 0 5 Y 10 Y Y**. In this example, the sweeps will be rejected when the amplitude from the 0-5Hz band exceeds 10uVs (Bad and Skipped channels are excluded).

**ARTREJFREQ\_EX** - Artifact Rejection based on frequency band amplitude, extended.  The **ARTREJFREQ\_EX** command is similar to the **ARTREJFREQ** command except that it allows you to specify which channels should be the artifact rejection channels, and the options to exclude bad or skipped channels are removed (since they are really not needed when you select the channels to include).

1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	double	Start reject frequency
3	double	Stop reject frequency
4	Boolean	Recompute
5	double	Maximum amplitude
6	list	List of artifact rejection channels

*Example.* **ARTREJFREQ\_EX REJCRITERIA 0 5 Y 10 { FP1 F7 T3 T5 FPZ FT7 FP2 F8 T4 T6 FT8 }**. In this example, the sweeps will be rejected when the amplitude from the 0-5Hz band exceeds 10uVs, using the indicated channels for artifact channels.

**\*BLINKNOISEREDUCTION** - Gradient/Blink Noise Reduction Routine.  This is one of the commands used with the Tool Box 2003 add-on software (to access it you must be under a maintenance contract; contact [sales@neuro.com](mailto:sales@neuro.com) or [techsup@neuro.com](mailto:techsup@neuro.com) for details). The **BLINKNOISEREDUCTION** command is intended for the removal of VEOG blink artifact from CNT files (see the **EDIT** manual for details). See also the **ARTCOR** command for an alternate (but similar) way to remove blink artifact.



1	string	Output file
2	double	Epoch start time (ms)
3	double	Epoch stop time (ms)
4	integer	Number of averages
5	Boolean	Use external trigger
6	double	Voltage threshold (uVs; ignored if #5 is true)
7	defined value	Trigger direction (Positive, Negative; ignored if #5 is true)
8	string	Artifact channel label (REQUIRED even if #5 is true)
9	int	Trigger code (ignored if #5 is false)
10	Boolean	Enable artifact rejection
11	double	Artifact min (uV)
12	double	Artifact max (uV)
13	Boolean	Insert events
14	int	Event code
15	Boolean	Enable bipolar reference
16	string	Bipolar reference channel label
17	Boolean	Enable high pass filter
18	double	High pass filter value

*Example.* BLINKNOISEREDUCTION

"c:\\ScanData\\demo\\corrected file.cnt" -50 300 15 N 250 P  
 "VEOG" X Y -100 100 Y 100 N "" N x. The output file is specified. The interval start and stop times create sweeps around the triggers from the VEOG channel where the positive voltage meets the 250 uV criterion. Artifact Rejection is enabled, using +/- 100 uV thresholds. Events with type codes of 100 are placed at the trigger points. The bipolar reference and high pass filter options are not enabled.

**CLEARART** - Clear Artifacts. \*\*\* The CLEARART command will reset all single-sweep accept/reject tags for all epochs in an EEG file to the accept state. This command is provided for compatibility with earlier DOS versions. The same result may be more easily implemented with the Clear All parameter in the ARTREJ command.

*Example.* The CLEARART command exists on a line by itself, and uses no parameters.

**\*EKGNOISEREDUCTION** - EKG Noise Reduction Routine. ☼ This is one of the commands used with the Tool Box 2003 add-on software (to access it you must be under a maintenance contract; contact sales@neuro.com or techsup@neuro.com for details). The EKGNOISEREDUCTION command is intended for the removal of ballistocardiogram and other EKG artifact from CNT files (see the EDIT manual for details).

1	string	Output file
2	double	Epoch start time (ms)
3	double	Epoch stop time (ms)
4	integer	Number of averages
5	Boolean	Use external trigger
6	double	Voltage threshold (uV; ignored if #5 is true)
7	defined value	Trigger direction (Positive, Negative; ignored if #5 is true)
8	string	Artifact channel label (ignored if #5 is true)
9	int	External trigger code (ignored if #5 is false)
10	Boolean	Enable artifact rejection
11	double	Artifact min (uV)
12	double	Artifact max (uV)
13	Boolean	Correct trigger channel
14	Boolean	Insert events
15	int	Event code
16	Boolean	Use external average
17	string	External AVG file (ignored if #16 is false)
18	Boolean	Enable bipolar reference
19	string	Bipolar reference channel label
20	Boolean	Enable high pass filter
21	double	High pass filter value

*Example.* EKGNOISEREDUCTION "c:\\ScanData\\demo\\corrected file.cnt" -50 300 15 N 100 P "EKG" x Y -100 100 N Y 100 N x N "" N x. The output file is specified. The interval start and stop times create sweeps around the triggers from the EKG channel where the positive voltage meets the 100 uV criterion. Artifact Rejection is enabled, using +/- 100 uV thresholds. Events with type codes of 100 are placed at the trigger points. The trigger channel is not corrected, and the rolling average is used (instead of using an existing AVG file). The bipolar reference and high pass filter options are not enabled.

**READREJ** - Read Rejected Sweeps File. \*\*\* The READREJ command reads a REJ file and rejects individual sweeps specified in the file. REJ files are created with the SAVEREJ command. This command can be used to recall a list of sweeps that have been previously rejected. For example, an EEG file could be visually edited for artifact, the list of rejected sweeps saved with the SAVEREJ command, and then recalled later with the READREJ command to reapply the manual rejection criteria. (The \*.rej file is an ASCII string of 0's and 1's, where 0=reject, and 1=accept).

1	string	Rejected sweeps file name (or "")
---	--------	-----------------------------------

*Example.*      `OPENFILE "c:\\ScanData\\Batch Examples\\p300.eeg"`  
                  `REVIEW`  
                  `SAVEREJ "c:\\ScanData\\Batch Examples\\rejsweeps.rej"`  
                  `CLEARART`  
                  `REVIEW`  
                  `READREJ "c:\\ScanData\\Batch Examples\\rejsweeps.rej"`

In this example, the file originalfile.eeg is retrieved. Single-sweeps are examined and tagged manually as accepted or rejected with the **REVIEW** command (click the button after reviewing the sweeps). The list of rejected sweeps are saved to the file rejsweeps.rej with the **SAVEREJ** command. All rejected sweeps are cleared with the **CLEARART** command, and you have the opportunity to verify that with the second **REVIEW** command. Finally, the rejected sweeps are restored with the **READREJ** command.

**SAVEREJ** - Save Rejected Sweeps File. \*\*\* The **SAVEREJ** command saves a REJ file to disk containing a table of sweeps that have been rejected in the EEG file. REJ files are read with the **READREJ** command. See **READREJ** above for an example.

1	string	Rejected sweep file name (or "")
---	--------	----------------------------------

*Example.* `SAVEREJ "c:\\ScanData\\Batch Examples\\rejectsweeps.rej"`.

**Averaging.** The primary command used to average single sweeps of either time or frequency domain files is the **AVERAGE** command, described below. This replaces the former **WAVEFORM** command. The former **VARIANCE** command is included now within the **AVERAGE** command (and is no longer recognized). The **GROUPAVG** command is used to average AVG files to form a group average.

**AVERAGE** - Average Sweeps. \*\*\* The **AVERAGE** command is used to average a sequence of single sweeps into an average file of all the selected sweeps. If you are using the Sort function, you will need to create the sort scheme first (see **CREATESORT**). If you are not using the sort function, use **NULL** in place of the sort name.

1	defined value	Domain (TIME, FREQUENCY)
2	Boolean	Compute standard deviation
3	Boolean	Compute SNR
4	string	SNR filename (or "")
5	defined value	Spectral scaling method (AMPLITUDE, POWER)
6	int	Spectral window length (Taper%)

7	defined value	Spectral window type (COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
8	string	Sort name (or "")
9	string	Output file name (or "")

*Example.* AVERAGE T N N "" 0 0 0 "MySort"  
"c:\\ScanData\\Batch Examples\\averaged file.avg". This will create an average of the sweeps using the MySort file (created beforehand with the CREATSORT command).

**AVERAGE\_EX** - \*\*\* This command is the same as the AVERAGE command except for the way in which the SNR is computed.

1	defined value	Domain (TIME, FREQUENCY)
2	Boolean	Compute standard deviation
3	defined value	Spectral scaling method (AMPLITUDE, POWER)
4	int	Spectral window length (Taper %)
5	defined value	Spectral window type (COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
6	defined value	Noise interval Type (PRESTIMINTERVAL, PERCENTILE, USERDEFINED)
7	float	SNR Noise Start
8	float	SNR Noise End
9	defined value	Signal interval Type (POSTSTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
10	float	SNR Signal Start
11	float	SNR Signal End
12	string	Sort name
13	string	Output file name

*Example.* AVERAGE\_EX TIME N AMPLITUDE 0 COSINE  
PRESTIMINTERVAL 0 0 POSTSTIMINTERVAL 0 0 ""  
{c:\\Scandata\\output file.avg. Sweeps from a previously retrieved epoched file are averaged in the time domain, using the prestimulus and poststimulus intervals for SNT estimation, with no sorting.

**GROUPAVG** - Group File Average. ~ This command is used to combine AVG files to form a group averaged AVG file. The formerly used **GROUP**, **INDIVIDUAL**, and **MINSWEEPS** commands have been discontinued. *Unlike almost all other data manipulation commands, this one does not use the current working file. In fact, there does not have to be any current working file for this transform to succeed.*

1	list	File names to be averaged ("" will fail)
2	defined value	Grouping method (GROUP, INDIVIDUAL)
3	Boolean	Compute variance
4	int	Minimum sweeps
5	Boolean	Exclude Bad channels
6	Boolean	Exclude Skipped channels
7	string	Output file (or "")

*Example.* GROUPAVG {{c:\ScanData\Batch Examples\file1.avg} {c:\ScanData\Batch Examples\file2.avg} {c:\ScanData\Batch Examples\file3.avg}} G Y 1 N N {c:\ScanData\Batch Examples\outputfile.avg}. Braces MUST be used with this command (not quotes). With braces, you MUST use single slashes in the path (\). The INDIVIDUAL option creates a group averaged file that is weighted by the number of sweeps in each file. This command is appropriate for combining similar recordings from the same subject that differ only in the number sweeps acquired.

*Hint:* You can use the **glob** command to avoid having to type all of the file names you want to include in the average (assuming they are in the same folder).

**AVGBANDS** - Modify Frequency Bands in AVG Files. (frequency domain AVG and COH files only). ~ This option formerly existed as the AVGBAND.exe utility program for the DOS version of EDIT. It is used to redefine the frequency bands that are exported to an ASCII file. The output is a text file with a .bnd extension. Repeat the label/start/stop sequence for as many bands as you want to create.

1	string	Output band file name
x	string	Band label
x	double	Band start frequency
x	double	Band stop frequency

*Example.* AVGBANDS "c:\ScanData\Batch Examples\freqband data.bnd" "Delta" 1 3.5 "Theta" 3.5 7 "Alpha" 8 13 "Beta" 14 25. The .bnd file will contain the specified bands and frequency limits (adjusted to precise frequency bin boundaries). A mode complete example might look like the following:

```
OPENFILE "c:\ScanData\Batch Examples\closed.avg"
AVGBANDS "c:\ScanData\Batch Examples\freqband data.bnd" "Delta"
1 3.5 "Theta" 3.5 7 "Alpha" 8 13 "Beta" 14 25
```

**Baseline Correction.** \*\*\*~ The principle command for performing Baseline Correction is BASECOR, described below. Related commands are

EXCLUDEFORBASECOR, RESETFORBASECOR and SETBASE (discontinued), also described below.

**BASECOR** - Baseline Correction. The baseline correct command allows you to modify the current DC offset of the waveform.

1	defined value	Interval type (FIRSTPOINT, PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start latency (if User Defined is selected)
3	double	Interval stop latency (if User Defined is selected)
4	Boolean	Exclude skipped channels
5	Boolean	Exclude bad channels
6	string	Output file name (EEG files only; or "")

*Example.* BASECOR PRE 0 0 N N "\$path\\basecorr.eeg". In this example, the prestimulus interval is used for baseline correction, and the Skipped and Bad channels are included. Note that the Output file parameter is used for EEG files only. With AVG files, you should omit the output file argument, and use SAVEAS to save the AVG file.

For EEG files:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\file1.eeg"
BASECOR PRE 0 0 N N "$path\\basefile1.eeg"
CLOSEFILE "file1.eeg"
```

For AVG files:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\file1.avg"
BASECOR PRE 0 0 N N
SAVEAS "$path\\basefile1.avg"
CLOSEFILE "file1.avg"
```

**BASECOR\_EX** - Include Channel(s) for Baseline Correction. This command differs from the BASECOR command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Interval type (FIRSTPOINT, PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start latency (only if 2 is USERDEFINED)

3	double	Interval stop latency (only if 2 is USERDEFINED)
4	list	List of electrodes to correct, or "All"
5	string	Output file (not needed for AVG files)

*Example.* BASECOR\_EX PRESTIM x x { FZ FCZ FPZ CZ PZ CPZ OZ } {}. Baseline correction will be applied using the prestimulus interval for the indicated electrodes only (no output file is needed for AVG files).

## **EXCLUDEFORBASECOR** - Exclude Channel(s) for Baseline Correction.

1...n	list	Electrode name(s)
-------	------	-------------------

*Example.* EXCLUDEFORBASECOR {Fp1 Fp2 F7 F8}. The indicated channels are excluded from baseline correction.

## **RESETFORBASECOR** - Reset Channel(s) for Baseline Correction. This command will reset all channels for baseline correction. It is typically used where only some channels had been set for baseline correction. It avoids having to reset each channel individually.

*Example.* The RESETFORBASECOR command is written on a line by itself.

## **CLEARALLMARKERS** - Clear All Markers. This command will remove markers that have been created with the Add Marker option (right mouse menu for Multiple Windows displays) or Peak Detection transform. All markers will be removed from all channels across all epochs (or the single epoch with AVG files).

*Example.* CLEARALLMARKERS. No additional parameters are used - all markers in the file will be removed.

## **CLEARMARKERS** - Clear Markers. This command is used to remove markers that have been created with the Add Marker option (right mouse menu for Multiple Windows displays) or Peak Detection transform. With EEG files, markers will be removed from the displayed sweep only (use CLEARALLMARKERS to remove all of them).

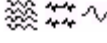
1	string	Selected channels, or All
---	--------	---------------------------

*Example.* CLEARMARKERS {CZ PZ OZ}. Markers will be deleted from the specified channels.


## **CLOSEALL** - Close All Files. The CLOSEALL command will close all files that you have opened. You will not be prompted to save modified data files. If the

transform you are using requires a separate output file, the **SAVEAS** command may be useful to save the file with a different name before closing. See also **OPENFILE** and **CLOSEFILE**.


*Example.* Place the **CLOSEALL** command on a line by itself to close all files. No parameters are used.

**CLOSEFILE** - Close File.  The **CLOSEFILE** command is used to close a specified file. Specify the file name and the extension without a path (e.g., viscpt.cnt). If you use **CLOSEFILE** you will not be prompted to save modified data files. If the transform you are using requires a separate output file, the **SAVEAS** command may be useful to save the file with a different name before closing. See also **OPENFILE** and **CLOSEALL**.

1	string	File name to close (no path; "" will fail))
---	--------	---

*Example.* **CLOSEFILE "newfile.avg"**. The designated file will be closed. Use **SAVEAS** first if you made changes to the file that you wish to save. The script below opens two files, arranges them vertically, and then closes the first one when you click the  button.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\file 1.avg"
OPENFILE "$path\\file 2.avg"
ARRANGEWINDOWS TILEV
PAUSE
CLOSEFILE "file 1.avg"
```

**COHERENCE** - Coherence.  The Coherence command will perform analyses on EEG files where the number of points must be a power of 2 (256, 512, etc.). Use the **SPLINEFIT** command if needed to get the correct number of points. (See the Coherence section of the EDIT manual for more details about the options with Coherence).

1	double	Computation of maximum frequency
2	defined value	Pair selection state (ALLPAIRS, MNTFILE)
3	string	MNT file name (ignored if param 2 is all pairs)
4	Boolean	Spectral covariance
5	Boolean	Retain mean
6	string	Output file name (or "")

*Example.* **COHERENCE 30 ALL "" N Y "c:\\ScanData\\Batch Examples\\outputfile.coh"**. In this example, coherence is computed up to 30Hz, on all pairs of electrodes, where the mean is Retained. (If you do not use an MNT file, you still need to add the empty string ("") in the 3rd field).



**SETCOHREF** - Set the Coherence Reference. ~ (COH files only) This command is used to set or change the Coherence Reference.

1	list	Electrode label
---	------	-----------------

*Example.* SETCOHREF "CZ". This sets CZ as the coherence reference.

**COHERENCE\_EX** - Coherence, Extended. \*\*\* The Coherence\_extended command is the same as the prior COHERENCE command, with the addition of a sorting option (new parameter 6).

1	double	Computation of maximum frequency
2	defined value	Pair selection state (ALLPAIRS, MNTFILE)
3	string	MNT file name (ignored if param 2 is all pairs)
4	Boolean	Spectral covariance
5	Boolean	Retain mean
6	string	Sort name
7	string	Output file name (or "")

*Example.* COHERENCE 30 ALL "" N Y "sort name"  
"c:\\ScanData\\Batch Examples\\outputfile.coh". In this example, coherence is computed using a sort (see CREATESORT).

**CREATESORT/DELETESORT** - Create/Delete a Sort File. ~\*\*\* The DOS version BATCH files required separate commands for various sorting parameters whenever you epoched or averaged a file. The previous DOS commands included ACCURACY, LATENCYRNG, MAXSWEEPS, READEVENT, RESPONSERNG, SEQUENCE, TRIALRNG, TYPERNG and VECTORSORT. These have all become obsolete in the current Tcl BATCH files.

The current software uses Sort "files". These are not actual files in the typical sense, but you may think of them as files without extensions. The "Sorts" are created and then applied in the AVERAGE, EPOCH, and other commands that use sorting.

To create a sort, use the CREATESORT command, followed by a sort name:

CREATESORT MySort

The sorting parameters are set in a subsequent series of commands (there must be a space before the dash, but not after it, as in -TrialEnabled):

MySort -TrialEnabled yes -TrialCriteria "1-3"

To use a sort, enter the sort name for the Sort Name parameter:

AVERAGE TIME no no "" 0 0 0 MySort result.avg

To delete a sort, use the **DELETESORT** command (this is not mandatory, Undeleted sorts are cleaned up automatically):

## DELETESORT MySort

In other words, the sorts are not saved - they must be recreated in the BATCH file, for use in that BATCH file.

The following is a list of sort parameters.

Parameter	type	defined values	default
-TrialEnabled	Boolean		NO
-TrialCriteria	string		""
-TypeEnabled	Boolean		NO
-TypeCriteria	string		""
-ResponseEnabled	Boolean		NO
-ResponseCriteria	string		""
-LatencyEnabled	Boolean		NO
-LatencyMin (ms)	double		0
-LatencyMax (ms)	double		0
-CorrectEnabled	Boolean		NO
-CorrectCriteria	defined value	(CORRECT INCORRECT BOTH NORESPONSE)	CORRECT
-SortOnEnabled	Boolean		NO
-SortOnCriteria	defined value	(EVEN ODD RANDOM)	EVEN
-MaxSweeps	int		-1
-SeedType	defined value	(CLOCK USERDEFINED)	CLOCK
-RandomSeed	int		0

When a new sort is created, all of the variables are set to the defaults listed above (all the sorting options are No, or FALSE, so the default sort will not exclude any epochs).

You may set one, many or all desired parameters on a single, or use multiple lines:

```
MySort -TrialEnabled YES -TrialCriteria "1-3"
or
MySort -TrialEnabled YES
MySort -TrialCriteria "1-3"
```

The two examples are equivalent.


The parameter and the value **MUST** be listed in pairs. If you have a line such as:

```
MySort -MaxSweeps
you will get an error (Invalid number of parameters).
```

For commands like **AVERAGE**, you may not always wish to use sorting. In such cases, just specify **NULL** in the place of the sort name:


**AVERAGE TIME no no "" 0 0 0 NULL result.avg**

This will average all of the epochs (i.e., no sorting is applied).

**CUTEPOCH** - Cut Epoch.  The Cut Epoch option will create a new AVG or EEG file with Start and Stop time points that are less than the original file. *The Start and End latencies MUST be within the original epoch interval.*

1	double	Start latency
2	double	End latency
3	string	Output file name (or "")

*Example.* **CUTEPOCH -100 500 "\\ScanData\\Batch Examples\\outputfile.avg"**. In this example, a new file (.avg or .eeg file) from -100 to 500 ms will be created.


**DCCORRECT** - DC Correction.  The DC Correction option will attempt to correct for DC offsets and DC drifting in continuous files recorded on a SynAmps in DC mode with a DC High Pass filter (see the EDIT manual for complete details).

1	double	Pre correction start time
2	double	Pre correction stop time
3	double	Post correction start time
4	double	Post correction stop time
5	Boolean	Perform DC drift correction
6	Boolean	Save drift waveforms
7	int	Polynomial order
8	Boolean	Exclude stimulus events
9	Boolean	Exclude response events
10	Boolean	Exclude keyboard events
11	double	Exclude interval start time
12	double	Exclude interval stop time
13	string	Drift file name (ignored if save drift is False)
14	Boolean	Use interval mode

*Example.* **DCCORRECT -1000 0 500 1500 Y N 3 Y Y Y 0 0 "" N**. The DC correction is performed using the options indicated.

The last parameter - Use Interval Mode - refers to the *Entire recording / Between corrections* options seen in point-and-click mode. In the DC correction process, one of the early steps is to remove the DC offset corrections. In longer files with gradual drifting, or shorter files with more severe drifting, this could result in some clipping of data toward the end of the file as the signals approach saturation.


The 14th parameter gives you the option to use the *Entire recording* (same as in versions of SCAN prior to 4.3), or to use the *Between corrections* option, or the "Interval Mode". This avoids (or minimizes) the potential clipping problem by analyzing the sections between the DC corrections independently, rather than using the entire file.

**DECIMATE** -  Use this option to decimate CNT files to a lower AD rate. Retrieve the CNT file, select the Decimate transform, and select a New Sample rate from the list of pull-down options. Enter an output file name, and click OK. A new CNT file will be created with the lower AD rate (and smaller file size). Prior to decimation, an IIR low pass filter is applied (passed four times) to correct for aliasing.

1	int	New AD Rate
2	string	Output file name


Example. DECIMATE 125 {C:\Scan Data\Demo Files\EEGs\closed decimate.cnt}. The new CNT file will be created.

Valid AD rates are only those rates that are lower submultiples of the AD rate in the data file. The submultiples are computed automatically. To determine what AD rates are valid, run the Decimate transform manually with the data file, and see what options are in the pull-down menu. Other files with the same AD rate will have the same decimation options.

**DELETEBADCHANNELS** - Delete Bad Channels.  The DELETEBADCHANNELS command is used to create a new continuous file with the "Bad" channels excluded. Use the SETCHANATTRIBUTE command (in the ACQUIRE and EDIT commands) to designate a channel as being Bad, if desired.


1	string	Output file name (or "")
---	--------	--------------------------

Example. (After retrieving a CNT file with Bad channels to be removed)  
DELETEBADCHANNELS "c:\ScanData\Batch Examples\NoBadChannels.cnt".

**DELETESWEEPS** - Delete Rejected Sweeps.  This command will create a new epoched file with the rejected sweeps removed. Use the ARTREJ command to reject the sweeps automatically, then create the new file, if desired.

1	string	Output file name (or "")
---	--------	--------------------------

Example. (After retrieving an EEG file with sweeps to be removed)  
DELETESWEEPS "c:\ScanData\Batch Examples\NoRejSweeps.eeg".

**Detrending (Linear Detrending).**  The principle command for applying the Linear Detrend transform is DETREND, described below. Related commands are EXCLUDEFORDETREND and RESETFORDETREND, also described below.

**DETREND** - Linear Detrend. This option is used to remove linear drifting components (such as, HEOG artifact) from EEG or AVG files.

- |   |               |   |
|---|---------------|---|
| 1 | defined value | Interval type (PRESTIMINTERVAL, ENTIRE INTERVAL, USERDEFINED) |
| 2 | double        | Interval start (ignored if parameter 1 is not "User defined") |
| 3 | double        | Interval end (ignored if parameter 1 is not "User defined")   |
| 4 | string        | Output file name (or "")                                      |

*Example.* DETREND ENT 0 0 "c:\ScanData\Batch Examples\detrended file.eeg". The channels will be detrended using the entire interval, and a new file is created.

**DETREND\_EX** - Include Channel(s) for Detrending. This command differs from the DETREND command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

- |   |               |  |
|---|---------------|--|
| 1 | defined value | interval type (PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED) |
| 2 | double        | Interval Start(ignored if parameter 1 is not USERDEFINED)    |
| 3 | double        | Interval End(ignored if parameter 1 is not USERDEFINED)      |
| 4 | list          | Channels to detrend  |
| 5 | string        | Output file name (or "")                                     |

*Example.* DETREND\_EX PRESTIMINTERVAL x x { F3 F4 C3 C4 P3 P4 F7 F8 T5 T6 CZ FZ PZ OZ} {C:\ScanData\Demo\Veps\output.eeg} . Linear detrending will be applied using the prestimulus interval for the selected channels.

**EXCLUDEFORDETREND** - Exclude Channel(s) for Linear Detrend. Use this command to exclude channels from Detrending.

- |       |      |                   |
|-------|------|-------------------|
| 1...n | list | Electrode name(s) |
|-------|------|-------------------|

*Example.* EXCLUDEFORDETREND {Fp1 Fp2 F7 F8}.

**RESETFORDETREND** - Reset Channel(s) for Linear Detrend. This command will reset all channels for linear detrending (so that none are excluded). It is typically used where only some channels had been set for baseline correction. It avoids having to reset each channel individually.

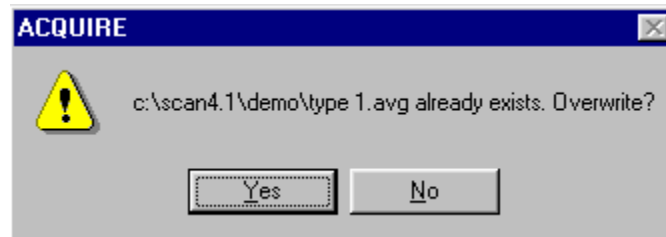
*Example.* The RESETFORDETREND command is written on a line by itself. No parameters are used.

**ENABLEOVERWRITEPROMPT** - Overwrite Existing File Prompt. This command is used to display or not display the message asking whether you wish to overwrite an existing data file. Disabling the command will allow the BATCH program to run without user intervention (be sure that you do not unintentionally overwrite files you want to keep).

1	Boolean	Enable overwrite prompt
---	---------	-------------------------


*Example.* ENABLEOVERWRITEPROMPT N. You will not see the normal "Overwrite" prompt (the file will be overwritten without a warning). For example, in the script below you will automatically see the "Overwrite" warning.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\type 1.avg"
SAVEAS "$path\\type 1.avg"
```



If you include ENABLEOVERWRITEPROMPT N prior to the SAVEAS command, the file will be overwritten without warning, and the script will continue without any user intervention.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\type 1.avg"
ENABLEOVERWRITEPROMPT N
SAVEAS "$path\\type 1.avg"
```

**EPOCH** - Epoch.  The EPOCH command will transform a continuous (\*.CNT) file into an epoched or single-sweep file (\*.EEG). See also CREATSORT.

1	defined value	Trigger mode (PORT_INTERNAL, NOTRIGGER, EVENTFILE)
2	string	Event file name (ignored unless #1 is EVENTFILE)

3	double	Start latency
4	double	Stop latency
5	Boolean	Response locked
6	Boolean	Reject epochs that overlap rejected blocks
7	Boolean	Include stimulus events
8	Boolean	Include keyboard events
9	Boolean	Include response pad events
10	string	Sort name
11	string	Output file

*Example.* EPOCH PORT "" -100 800 N Y Y N N "Sortfile"  
"\$path\\epoched file.eeg". Epochs from -100 to 800ms will be created using the indicated "Sortfile". The sort file is created with the CREATSORT command. (It may be necessary to merge the DAT file - MERGEEVT - from STIM beforehand to have access to the behavioral information).

**EPOCH\_EX** - Epoch, Extended. ⚠ The EPOCH\_EX command differs from the original EPOCH command in that it will accept an event file that is seconds (rather than byte or point offsets).

1	defined value	Trigger mode (PORT_INTERNAL, NOTRIGGER, EVENTFILE)
2	string	Event file name (ignored unless #1 is EVENTFILE)
3	Boolean	Event file is in seconds (ignored unless #1 is EVENTFILE)
4	double	Start latency
5	double	Stop latency
6	Boolean	Response locked
7	Boolean	Reject epochs that overlap rejected blocks
8	Boolean	Include stimulus events
9	Boolean	Include keyboard events
10	Boolean	Include response pad events
11	string	Sort name
12	string	Output file

*Example.* EPOCH PORT "" x -100 800 N Y Y N N "Sortfile"  
"\$path\\epoched file.eeg". Epochs from -100 to 800ms will be created using the indicated "Sortfile". The sort file is created with the CREATSORT command. (It may be necessary to merge the DAT file - MERGEEVT - from STIM beforehand to have access to the behavioral information).

**ERBP** - Event Related Band Power. ⚠ This transform computes power (or amplitude) of induced (and/or evoked) event-related EEG activity in a centered frequency band as a function of time. The options that are available for some parameters are dependent on the selections in some other parameters. For example, if you select POWER for #1, the AMPLITUDE option is ignored in #15. If #10 is not enabled, #11 and #12 will be

ignored. There are several other examples. We strongly encourage you to use the ERBP window in Point & Click mode to set up the transform, and then transfer those settings to your BATCH file (or use the Auto-write feature to have the command created automatically). See also CREATSORT.

1	defined value	Method (POWER, AMPLITUDE)
2	Boolean	Enable bandpass
3	defined value	Filter Mode(ZEROPHASESHIFT, ANALOGSIMULATION)
4	double	Center frequency (Hz)
5	double	Half bandwidth (Hz)
6	int	Bandpass attenuation (dB)
7	Boolean	Enable Envelope
8	Boolean	Warm up from right
9	defined value	Phase locking (INDUCED, EVOKED, BOTH)
10	Boolean	Enable reference interval
11	double	Reference start (ms's)
12	double	Reference stop (ms's)
13	double	Trim left (ms's)
14	double	Trim right (ms's)
15	defined value	Scaling (POWER, AMPLITUDE, PERCENT, Z)
16	Boolean	Enable Averaging window
17	double	Averaging window (ms's; ignored if #16 is False)
18	defined value	Window treatment (COLLAPSE, SMOOTH)
19	string	Sort name
20	string	Output file

*Example.* ERBP POW Y ZERO 10 2 24 N N BOTH Y -100 0 100 100 PER Y 100 COLL "" "\$path\ERBP output.avg". This will perform the "traditional" event-related desynchronization.

**ERCOH** - Event Related Coherence. \*\*\* ERCoh is computed from epoched EEG data using the coherence formulas already given in the EDIT manual in the Coherence section. However, in this case, the frequency of interest is preselected, and the results are a function of time with respect to the event at time zero (refer to the EDIT manual for more details). The options that are available for some parameters below are dependent on the selections in some other parameters. For example, if you select COHERENCE in #1, the EVOKED option is ignored in #6. If you select COVARIANCE in #1, EVOKED and BOTH are ignored in #6. If you select CROSSPRODUCT in #1, INDUCED is ignored in #6. It would be a good idea to use the ERCOH window in Point & Click mode to set up the transform, and then transfer those settings to your BATCH file. See also CREATSORT.

1	defined value	Type (COHERENCE, COVARIANCE, CROSSPRODUCT)
2	double	Center frequency (Hz)



3	double	Half bandwidth (Hz)
4	int	Bandpass attenuation (dB)
5	Boolean	Warm up from right
6	defined value	Phase locking (INDUCED, EVOKED, BOTH)
7	double	Trim left
8	double	Trim right
9	defined value	Pair selection (ALLPAIRS, MNTFILE)
10	string	MNT file
11	string	Sort name
12	string	Output file (.coh extension)

*Example.* ERCOH COH 5 2 24 N IND 100 100 ALL "" ""  
"\$path\ERCOH file.coh". This is a basic illustration of ERCoh where the mean activity (phase locked) is removed, and the results are normalized so that the final results are numbers between 0 and 1. The center frequency of interest is 5Hz, and ERCoh was computed for all pairs of channels. The sweeps have been trimmed by 100ms on each side.

**Exporting/Importing ASCII Files.** These commands are used for exporting files to and importing files from ASCII.

**EXPORTAVG** - Export AVG File to ASCII. \*\*\* This command allows you to export averaged time or frequency domain data to an ASCII file.

1	string	Output file name (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include electrode labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include standard deviation
8	Boolean	Include Bad channels
9	Boolean	Include Skipped channels

*Example.* EXPORTAVG "c:\ScanData\Batch Examples\outputascii.dat" POINTS Y Y N N N Y Y. The ASCII file will be created where the columns contain the data points for each channel, and the header, labels, and all channels will be included. The X and Y units will not be included.

Below is an example demonstrating how several files can be exported in a single BATCH file.

```
set path "c:\ScanData\Batch Examples\"
```

```

set filelist {1BKE_R2 1BKF_C2 1CDA_R2 1CDB_C2 1DKA_R2
1DKB_C2 1DLA_C2 1DLB_R2 1GCC_C1 1GCD_R1 1JRA_C1
1JRB_R1 1LTA_C1 1LTB_R1 1RCA_R2 1RCB_C2 1SDA_R2
1SDB_C2 1TAC_R2 1TAD_C2 1UUE_C1 1UUF_R1 1YSA_R2
1YSB_C2
}

```

```

ENABLEOVERWRITEPROMPT FALSE

```

```

foreach element $filelist {

```

```

# OPENS AND EXPORTS A LIST OF *.AVG (List is 1 file or more in length)
# FILES AS A FORMATTED ASCII FILE

```

```


    set file $path$element
    OPENFILE "$file.avg"
    SETSKIP "HEOG REF VEOGL" ON
    SAVEAS "$file.avg"
    EXPORTAVG $file.dat POINTS 1 1 1 1 0 0 0
    CLOSEALL
}

```

**EXPORTAVG\_EX** - Export AVG files, Extended. \*\*\* This command allows you to export AVG files with the additional options to include Data Labels and Maximum Resolution (to at least 8 decimal places). The Data Label is the line: [Average Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also. Enabling the last parameter will append the new data to an existing data file with the same name (if the file already exists).


1	string	Output file name (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include electrode labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include standard deviation
8	Boolean	Include Bad channels
9	Boolean	Include Skipped channels
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	list	Channels to be exported (or "all")

*Example.* EXPORTAVG\_EX {C:\ScanData\Demo\output.dat} P Y Y Y Y N Y Y Y N N {ALL}. The file will be exported where rows=points, with the header, electrode labels, X and Y units, Bad and Skip channels, and the data label. The default resolution is used (up to 4 decimal places). All channels will be exported.

**EXPORTCNT** - Export CNT File to ASCII.  This command allows you to export a CNT file to an ASCII file. CNT files are exported using the "rows = points" format only.

1	string	Output file (or "")
2	Boolean	Include header
3	Boolean	Include labels
4	Boolean	Include X units
5	Boolean	Include Y units
6	Boolean	Include bad channels
7	Boolean	Include skipped channels

*Example.* EXPORTCNT "c:\ScanData\Batch Examples\cntfile.dat" Y Y N N N N. The ASCII file will be created where the columns contain the data points for each channel.

**EXPORTCNT\_EX** - Export CNT File to ASCII, Extended.  This command allows you to export a CNT file to an ASCII file, with the additional options to include the Data Labels and to use Maximum Resolution. The Data Label is the line: [Continuous Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also. Enabling the last parameter will append the new data to an existing data file with the same name (if the file already exists).

1	string	Output file
2	Boolean	Include header
3	Boolean	Include labels
4	Boolean	Include X units
5	Boolean	Include Y units
6	Boolean	Include bad channels
7	Boolean	Include skipped channels
8	Boolean	Include data labels
9	Boolean	Use maximum resolution
10	Boolean	Append if file exists
11	list	Channels to be exported (or "all")

*Example.* EXPORTCNT\_EX "c:\ScanData\Batch Examples\cntfile.dat" Y Y N N N N Y Y N {ALL}. The ASCII file will be created where the columns contain the data points for each

channel. The data labels are included and the values are written with maximum resolution (to at least eight decimal places). All channels will be exported.

**EXPORTCOH\_EX** - Export COH files, Extended. ~ This command allows you to export COH files to ASCII. If you want to limit the pairs that are exported, you should use an MNT file when you execute the COHERENCE transform.

1	string	Output file (or "")
2	defined value	Method (POINTS, PAIRS)
3	Boolean	Include header
4	Boolean	Use maximum resolution
5	Boolean	Append if file exists

Example. EXPORTCOH\_EX "c:\\ScanData\\Batch Examples\\cohfile.dat" PAIRS Y N N. The COH file will be exported to ASCII, where the rows are the pairs of electrodes. The header is included, normal resolution is used (up to 4 decimal places), and if there is an existing file with the same name, it will be overwritten.

**EXPORTEDF** - Export CNT File in European Data Format (EDF). ☼ This command allows you to export a CNT file in European Data Format.

1	string	Output file (or "")
2	Boolean	Include bad channels
3	Boolean	Include skipped channels

Example. EXPORTEDF "c:\\ScanData\\demo\\EDF file.rec" Y Y. The CNT file is exported in EDF, with the bad and skipped channels included.

**EXPORTEEG** - Export EEG File to ASCII. ☼☼ This command allows you to export a EEG file to an ASCII file.

1	string	Output file (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include bad channels
8	Boolean	Include skipped channels
9	Boolean	Include epoch headers

Example. EXPORTEEG "c:\\ScanData\\Batch Examples\\eegfile.dat" P Y Y N N N N Y. The ASCII file will be

created where the columns contain the data points for each channel, excluding Bad and Skip channels, with the headers for each epoch.

**EXPORTEEG\_EX** - Export EEG File to ASCII, Extended. ✖✖ This command allows you to export a EEG file to an ASCII file, with the additional options to include the Data Labels and to use Maximum Resolution (maximum number of decimal places). The Data Label is the line: [Epoched Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also. Enabling the last parameter will append the new data to an existing data file with the same name (if the file already exists).

1	string	Output file (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include bad channels
8	Boolean	Include skipped channels
9	Boolean	Include epoch headers
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	list	Channels to be exported (or "all")

*Example.* EXPORTEEG\_EX "c:\\ScanData\\Batch Examples\\eegfile.dat" P Y Y N N N N Y Y N N {ALL}. The ASCII file will be created where the columns contain the data points for each channel, excluding Bad and Skip channels, with the headers for each epoch. The data labels are included, and the resolution is set for the default number of decimal places (4). All channels will be exported.

**IMPORTAVG** - Import AVG File from ASCII. ~✓ This command allows you to import average waveform/spectrum data that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the AVG type. If the information listed in the parameters below is already contained in the ASCII file header, then the parameters will be ignored. If the information is not in the header, then supply it using the parameters. You may wish to use the READPOS command (created from an existing, matching AVG file using the Overall Layout screen) to position the electrodes automatically (otherwise they will be displayed in a grid format).

1	string	File name (or "")
2	defined value	Method (POINTS, ELECTRODES)
3	int	Number of channels

4	double	Acquisition rate
5	double	X min
6	int	Number of points
7	int	Number of sweeps used to create the average
8	Boolean	Frequency domain

*Example.* IMPORTAVG "c:\\ScanData\\Batch  
Examples\\avgascii.dat" P 30 1000 -50 550 200 N. The ASCII  
data will be imported.

**IMPORTCNT** - Import CNT File from ASCII. \*\*\* This command allows you to import continuous data that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the CNT type. Files are imported in the "rows = points" format only. The last parameter is used to specify Neuroscan or Nihon Koden data files (files are assumed to be Neuroscan CNT files by default - you may leave the 4th parameter blank if you are importing Neuroscan CNT files).


1	string	File name (or "")
2	int	Number of channels
3	double	Acquisition rate
4(opt)	defined value	Company (NEUROSCAN, NK)

*Example.* IMPORTCNT "c:\\ScanData\\Batch  
Examples\\cntascii.dat" 30 1000. The Neuroscan ASCII data file will  
be imported.

**IMPORTEEG** - Import EEG File from ASCII. \*\*\* This command allows you to import epoched data (sweeps) that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the EEG type.


1	string	File name (or "")
2	defined value	Method (POINTS, ELECTRODES)
3	int	Number of channels
4	double	Acquisition rate
5	double	X min
6	int	Number of points
7	int	Number of sweeps
8	Boolean	Frequency domain

*Example.* IMPORTEEG "c:\\ScanData\\Batch  
Examples\\eegascii.dat" P 30 1000 -50 550 200 N. The ASCII  
data will be imported.

**EXTRACT** - Extract Channels.  The **EXTRACT** command creates a file containing data from selected channels from the current AVG or EEG file. The electrode arguments must match electrode labels in the current file. See also **APPEND**.

- |   |        |                          |
|---|--------|--------------------------|
| 1 | list   | Channels to be included  |
| 2 | string | Output file name (or "") |

*Example.* **EXTRACT {FZ CZ PZ OZ} {c:\ScanData\fewchannels.avg}**. Braces **MUST** be used with this command (not quotes). With braces, you **MUST** use single slashes in the path (\). A new file will be created containing only the data from the selected channels.

**Filtering data files.**  The principle command for filtering data files is **FILTER**. Related commands are the **EXCLUDEFORFILTER** and **RESETFORFILTER** commands. These are described below. The **SETFILTRECTIFY** command is now contained within the **FILTER** command.

**FILTER** - Filter. Use this command to perform a variety of filtering options on your data file.

- |         |               |  |
|---------|---------------|--|
| 1       | defined value | Filter type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)  |
| 2       | defined value | Filter mode (ZEROPHASESHIFT, ANALOGSIMULATION)   |
| 3       | float         | High pass cutoff   |
| 4       | int           | High pass dBs (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48) |
| 5       | float         | Low pass cutoff  |
| 6       | int           | Low pass dBs (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)  |
| 7       | float         | Notch start frequency  |
| 8       | float         | Notch stop frequency   |
| 9       | int           | Notch poles  |
| 10      | Boolean       | Rectify  |
| 11(opt) | string        | Output file name (or ""); CNT and EEG files only   |

*Example.* **FILTER BANDPASS ZERO 1.5 24 50 24 0 0 0 N "c:\ScanData\Batch Examples\filteredfile.eeg"**. Bandpass, zero-phase shift filtering (no latency shifts) is performed with low pass filter setting of 50Hz (24dB) and a high pass filter of 1.5Hz (24dB), with the creation of a new filtered data file.

**FILTER\_EX** - Include Channel(s) for Filtering. This command differs from the **FILTER** command in that it allows you to list the electrodes that you want to include (or "All"). See the **"\_EX" Commands** section above. The "Auto-write"

feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Filter Type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
2	defined value	Filter Mode (ZEROPHASESHIFT, ANALOGSIMULATION)
3	float	High Pass cutoff
4	int	High Pass attenuation (db) (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)
5	float	Low Pass cutoff
6	int	Low Pass attenuation (db) (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)
7	float	Notch start frequency
8	float	Notch stop frequency
9	int	Notch attenuation (db)
10	Boolean	Rectify
11	defined value	Filter class (FIR, IIR)
12	list	List of electrodes to filter or "All"
13(opt)	string	Output file name (or ""); CNT and EEG files only

*Example.* FILTER\_EX BAND ZERO 1.0 24 30.0 24 x x x N  
FIR { FP1 F3 F7 FZ FPZ FT7 FP2 F4 F8 FT8 VEOG } {}.  
FIR Filtering will be applied using the indicated parameters to selected channels.

**EXCLUDEFORFILTER** - Exclude Channel(s) for Filtering. Use this command to exclude channels from Filtering.

1...n	list	Electrode name(s)
-------	------	-------------------

*Example.* EXCLUDEFORFILTER {Fp1 Fp2 F7 F8}.

**RESETFORFILTER** - Reset Channel(s) for Filtering. This command will reset all channels for linear detrending. It is typically used where only some channels had been set for baseline correction. It avoids having to reset each channel individually.

*Example.* The RESETFORFILTER command is written on a line by itself. No parameters are used.

**FSP** - Fsp Average. \*\*\* The Fsp Average command performs a signal to noise (SNR) computation on epoched files as a means for helping to determine when a signal has



emerged with statistical significance from the noise background (refer to the EDIT manual for complete details). See also **CREATESORT**.

1	defined value	Terminate method (NONE, FVALUE, NOISELEVEL, BOTH)
2	double	F value
3	double	Noise level
4	double	Single point position
5	int	Sweeps per block
6	double	Window start
7	double	Window stop
8	string	Sort name
9	string	Output file name

*Example.* FSP FVALUE 3.1 0.1 10 200 -2 15 NULL  
 "c:\\ScanData\\Batch Examples\\Fspoutput.avg". This line performs a basic FSP calculation, with no sorting.

**FSP\_EX** - Include Channel(s) for Fsp Averaging. This command differs from the FSP command in that it allows you to list the electrodes that you want to include for the termination channels (or "All"). See the **"\_EX" Commands** section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Terminate Method (NONE, FVALUE, NOISELEVEL, BOTH)
2	double	F Value,
3	double	Noise Level,
4	double	Single Point Position
5	int	Sweeps Per Block
6	double	Window Start
7	double	Window Stop
8	string	Sort Name
9	list	List of terminate electrodes or "All"
10	string	Output file

*Example.* FSP\_EX BOTH 3.10 0.10 10.0 200 0.0 20.0 NULL { FP1 FP2 VEOG} {C:\\ScanData\\Demo\\Visual Attention\\erase me.eeg}. Fsp averaging will be conducted using F value and Noise level criteria, with selected channels as the termination channels, and the other parameters as indicated.

**Get/Set commands.** There are several new "GET" commands, and many of these return a value from the data file. Several of these have related "SET" commands, used to change settings or enter new values. Some of the commands are useful for, among other things, accessing and modifying the event table (the event information contained at the end of CNT files).

**GETCOMP/DELETEALLCOMP** - Get/Delete Comparison Files. Use the GETCOMP command to open a comparison AVG file and superimpose it on a previously retrieved working file. (The files must match in terms of electrode labels, number of channels, number of points, and latency/frequency start/stop times). Use the DELETEALLCOMP command (no parameters) to delete the comparison file(s).

1                      string                      Comparison file name (or "")

*Example.* GETCOMP "c:\\ScanData\\Batch Examples\\myfile.avg".

**GETCURRENTEPOCHINDEX/SETCURRENTEPOCHINDEX** - Get/Set Current Epoch Index. These commands will return the sweep number of the currently displayed epoch. GETCURRENTEPOCHINDEX uses no parameters. SETCURRENTEPOCHINDEX uses the single sweep number parameter and displays the specified sweep.

**Return.** The sweep number of the currently displayed epoch is returned.

1            int            Sweep number of the epoch to display (zero-based index)

*Example.* GETCURRENTEPOCHINDEX. The command is entered on a line by itself, and no parameters are used. To return and display the current epoch number, use the INSTRUCT command, as shown below.

```
set index [GETCURRENTEPOCHINDEX]
INSTRUCT $index
```

The current sweep number will be displayed. *Reminder: the BATCH commands use a zero-based index, meaning that, in this case, the first sweep is sweep 0, not sweep 1. The returned value will be one less than the value seen in the Status box in EDIT.*



*Example.* SETCURRENTEPOCHINDEX 37. The 38th epoch will be displayed (the index is zero-based, so specifying the 37th sweep will display the actual 38th sweep).

**GETEPOCHCOUNT** - Get Epoch Count. This command will return the number of sweeps or epochs in an EEG file.

**Return.** The number of sweeps in the file is returned.

*Example.* GETEPOCHCOUNT appears on a line by itself, and uses no parameters. To see the returned value, use the INSTRUCT command, as follows.

```
set index [GETEPOCHCOUNT]
INSTRUCT $index
```

The total number of sweeps in the file will be displayed.



### **GETEPOCHINFO / SETEPOCHINFO** - Get/Set Epoch Information. \*\*\*

These commands will return information from EEG files regarding specified epochs (GETEPOCHINFO), or to make changes for specified epochs (SETEPOCHINFO).

**Return.** The specified information is returned. The returned values may be interpreted using the list below.

#### **GETEPOCHINFO**

- |   |               |   |
|---|---------------|---|
| 1 | int           | Epoch number (zero-based index)                                     |
| 2 | defined value | Parameter (-TrialType, -Accept, -Correct, -ReactionTime, -Response) |

*Example.* GETEPOCHINFO 45 -A. The program will return the "Accept" field for the 46th sweep. (Reminder: with a zero-based index, the first sweep is really sweep 0, although it is displayed as sweep 1 in the Status Box). To see the returned value, use a script similar to the following:

```
SET PATH "C:\\SCANDATA\\BATCH EXAMPLES"
OPENFILE "$PATH\\VISCPT.EEG"
set accept [GETEPOCHINFO 45 -A]
INSTRUCT $accept
CLOSEALL
```

The returned value will appear as follows.



The returned numbers mean the following (depending on what type of information is requested):

-TrialType	Returns the stimulus type code number
-Accept	0 = Reject, 1 = Accept
-Correct	0 = Incorrect, 1 = Correct
-ReactionTime	Returns the latency of the response
-Response	Returns the response type code number


### SETEPOCHINFO

1	int	Epoch number (zero-based index)
2	defined value	Parameter (-TrialType, -Accept, -Correct, -ReactionTime, -Response)
3	varies	New value to replace the existing one

*Example.* SETEPOCHINFO 45 -Accept 0. This changes the Accept field from 1 to 0. To see the returned values, use a script similar to the one below. In this example, the reaction time is changed as well.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\viscpt.eeg"
set accuracy [GETEPOCHINFO 45 -Accept]
set latency [GETEPOCHINFO 45 -React]
INSTRUCT $accuracy
INSTRUCT $latency
SETEPOCHINFO 45 -Accept 0
SETEPOCHINFO 45 -React 444
set newaccuracy [GETEPOCHINFO 45 -Accept]
set newlatency [GETEPOCHINFO 45 -React]
INSTRUCT $newaccuracy
INSTRUCT $newlatency
SAVEAS "$path\\newviscpt.eeg"
CLOSEALL
```

The above script changes the Accept field from 1 to 0, the reaction time from 402 to 444, displays the old and new values, and writes a new file with the changes.

**GETEVENTCOUNT** - Get number of events.  This command is used with CNT files only, and will return the total number of events in the file.

**Return.** The total number of events is returned.

*Example.* GETEVENTCOUNT. This command uses no parameters. To see the returned number of events, use the INSTRUCT command, as follows.

```
set evcount [GETEVENTCOUNT]
INSTRUCT $evcount
```

The returned information will be displayed.



## **GETEVENTINFO / SETEVENTINFO** - Get/Set Event Information.

These command will return information from the CNT file (only) event table (GETEVENT-INFO), or can be used to modify information in the event table (SETEVENTINFO).

**Return.** The returned value may be a number or a text string. See the list below to interpret the returned value.

### **GETEVENTINFO**

- |   |               |  |
|---|---------------|--|
| 1 | int           | Event number (zero-based index)  |
| 2 | defined value | Parameter (EventType, Offset, ResponseLatency, Accuracy, KeyboardCode, KeypadCode, StimulusCode) |

Depending upon which Parameter you selected for #2, the program will return differing information. The following translates the meaning of the returned codes.

- |                  |   |
|------------------|---|
| -EventType       | Stimulus, Keypad, Rejected sweep, Accepted sweep, Keyboard, DC Correction, and Segment (Stop/Start event) |
| -Offset          | Number of points since the beginning of the file.   |
| -ResponseLatency | The response latency in ms.   |
| -Accuracy        | No response, Incorrect, Correct   |
| -KeyboardCode    | Function key number (2-11)  |
| -KeypadCode      | Number of response pad button pressed (1-4)   |
| -StimulusCode    | The stimulus type code number (1-255)   |

*Example.* GETEVENTINFO 123 -Stim. The program will return the stimulus type code found for the 122nd event. (Reminder: the zero-based index assumes the first sweep is sweep 0, not sweep 1). A sample script is:

```
OPENFILE "c:\\ScanData\\Batch Examples\\viscpt.cnt"
set number [GETEVENTINFO 123 -Stim]
INSTRUCT $number
```

This returns:



### SETEVENTINFO

- |   |               |  |
|---|---------------|--|
| 1 | int           | Event number (zero-based index)  |
| 2 | defined value | Parameter (EventType, Offset, ResponseLatency, Accuracy, KeyboardCode, KeypadCode, StimulusCode) |
| 3 | varies        | Based on list below:   |

- |                  |               |   |
|------------------|---------------|---|
| -Offset          | int           | Number of points since the beginning of the file. |
| -ResponseLatency | double        | The response latency in ms.                       |
| -Accuracy        | defined value | No response, Incorrect, Correct                   |
| -KeyboardCode    | int           | Function key number (2-11)                        |
| -KeypadCode      | int           | Number of response pad button pressed (1-4)       |
| -StimulusCode    | int           | The stimulus type code number (1-255)             |

*Example.* SETEVENTINFO 12 -Keyp 3. In this example, a response pad type code of 3 is inserted into the 12th keypad value.

The following BATCH file examples demonstrate the GETEVENTINFO and SETEVENTINFO commands. The first shows how to replace a single stimulus type code number.


```
SET PATH "C:\\SCANDATA\\BATCH EXAMPLES"
OPENFILE "$PATH\\VISCPT.CNT"
set number [GETEVENTINFO 100 -Stim]
INSTRUCT $number
SETEVENTINFO 100 -Stim 10
set newnumber [GETEVENTINFO 100 -Stim]
INSTRUCT $newnumber
SAVEAS "$path\\newviscpt.cnt"
CLOSEALL
```

The value of the 100th stimulus value is returned (displayed), then changed to a new value. The new value is obtained and displayed, and the modified file is saved. The next example demonstrates how to replace a variety of arbitrary values.

```

OPENFILE {c:\ScanData\Batch Examples\AllEvents.cnt}
ARRANGEWINDOWS TILEH
set eventcount [GETEVENTCOUNT]
SAVEEVENT Y Y Y Y Y Y {c:\ScanData\Batch Examples\Before.EV2}
# This script demonstrates which parameters can be set on which type of
event.
for {set index 0} {$index < $eventcount} {incr index} {
    set type [GETEVENTINFO $index -EventType]
    set offset [GETEVENTINFO $index -Offset]
    set latency [GETEVENTINFO $index -ResponseLatency]
    set accuracy [GETEVENTINFO $index -Accuracy]
    set keyboard [GETEVENTINFO $index -KeyboardCode]
    set keypad [GETEVENTINFO $index -KeypadCode]
    set stim [GETEVENTINFO $index -StimulusCode]
# reset parameters to some arbitrary values
    switch $type \
        STIMULUS {SETEVENTINFO $index -Offset [expr $offset+100]
ResponseLatency 2.2 -Accuracy Correct -StimulusCode 88 } \
        KEYPAD {SETEVENTINFO $index -Offset [expr $offset-100]
-KeypadCode 4} \
        REJECT {SETEVENTINFO $index -Offset [expr $offset+500]} \
        ACCEPT {SETEVENTINFO $index -Offset [expr $offset-500]} \
        KEYBOARD {SETEVENTINFO $index -Offset [expr $offset-100]
-KeyboardCode 11} \
        DCCORRECTION {SETEVENTINFO $index -Offset [expr
$offset+1000]} \
        SEGMENT {SETEVENTINFO $index -Offset [expr $offset-1000]}
    }
SAVEEVENT Y Y Y Y Y Y {c:\ScanData\Batch Examples\After.EV2}

```

**GETMARKERCOUNT** - Get Marker Index Count.  This command returns the number of markers at a selected channel, and at a selected sweep. With AVG files, the 2nd parameter is ignored.

**Return.** The number of Markers is returned.

1	string	Channel label
2	int	Epoch index (zero-based; ignored for AVG files)

*Example.* GETMARKERCOUNT "OZ" 50. The number of markers in the 50th sweep (zero-based) at the OZ channel is returned. A sample sequence is as follows:

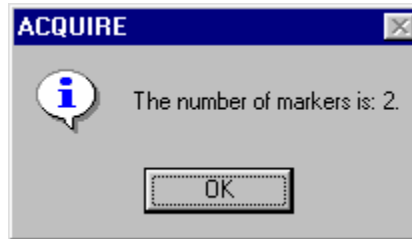
```


OPENFILE "$path\p300.eeg"
PEAKDETECTION "P300 Peak" Y 200 400 MAX Y "Pz" Y Y

```

```
PEAKDETECTION "N1 Peak" Y 80 120 MIN Y "Pz" Y Y
set marker [GETMARKERCOUNT "OZ" 50]
INSTRUCT "The numbers of markers is: $marker." OK
```

In this example, a "2" is returned.



**GETMARKERINFO** - Get Marker Information.  The GETMARKERINFO command will return specified information about a specified Marker, from a specified sweep. In a file with multiple markers, the number of the marker is generally determined by the order that the marker was placed - not in the order that they appear on the x-axis (see the example below). The program will return "Max" even if you had used Latency, Amplitude, or Both. Therefore, it is a good idea to verify the label of the marker you select to be sure it is the one you intend. With AVG files, the 4th parameter is ignored.

**Return.** The information that is returned depends upon what you request.

1	string	Channel label
2	int	Marker index (zero-based)
3	defined value	Parameter (XVALUE, YVALUE, LABEL)
4	int	Epoch index (ignored for AVG files)

*Example.* GETMARKERINFO "OZ" 1 XVALUE 50. The millisecond value for the zero-based "1" marker, from the OZ channel, from the zero-based 50th sweep, is returned.

In the following example, we are placing the P300 peak marker first, followed by the N1 peak second. The GETMARKERINFO command is used to return the latency of the "0th" marker (the P300 marker, since it was placed first), from the OZ channel, from the 50th sweep (zero-based sweep; i.e., the 51st sweep as counted in the file). Then, we are asking for the "Label" that was used for the same marker, channel, and sweep.

```
OPENFILE "$path\p300.eeg"
PEAKDETECTION "P300 Peak" Y 200 400 MAX Y "Pz" Y Y
PEAKDETECTION "N1 Peak" Y 80 120 MIN Y "Pz" Y Y
set latency [GETMARKERINFO "OZ" 0 XVALUE 50]
INSTRUCT "The millisecond value is: $latency."
set label [GETMARKERINFO "OZ" 0 LABEL 50]
```




INSTRUCT "The Label is: \$label."

The first value returned is:



The second returned value is:



**GETPOINTDATA** - Get Data Points.  This command will return the actual data points in the specified range and sweep number (if EEG file). The command is used with EEG and AVG files only.

**GETPOINTDATA:**

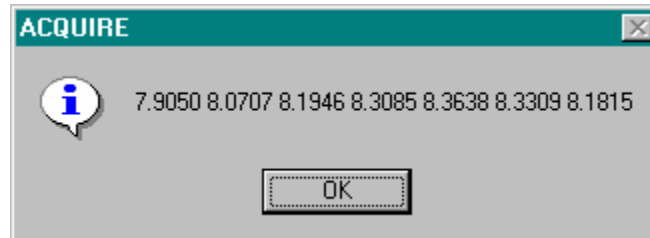
1	int	Start points (zero-based index)
2	int	End points (zero-based index)
3	string	Electrode label
4 (EEG files only)	int	Epoch number (zero-based index)


**Return.** The data point values within the specified range and epoch are returned.

*Example.* GETPOINTDATA 130 170 "OZ". The 130-170th data points, from the OZ electrode, are displayed. The last argument was omitted (an AVG file was opened, rather than an EEG file). If you know the AD rate and the starting point of the sweep, you can calculate the Start and End index numbers, or, you can use the LATENCYTOPOINT command to return them. Let's say you want to see the data points from 97-104ms. The example below shows how to obtain and display the selected data points, using LATENCYTOPOINT.

```
set path "c:\\ScanData\\Batch Examples"  
OPENFILE "$path\\vcp.avg"  
set start [LATENCYTOPOINT 97]  
set stop [LATENCYTOPOINT 104]
```


```
set data [GETPOINTDATA $start $stop {OZ}]
INSTRUCT $data
CLOSEALL
```



**GFP** - Global Field Power.  The GFP command will compute the global field power waveform and compute a common average reference for all electrodes, excluding Bad and Skip channels (refer to the EDIT manual for details). Use with CNT, EEG or AVG files. To create a reference using selected channels, please see the **REFER** command.

1	Boolean	Compute GFP channel
2	Boolean	Include reference electrode
3	Boolean	Compute field Z scores
4	string	Output file name (or "")

*Example.* GFP Y Y N "c:\\ScanData\\Batch Examples\\gfpfile.avg". GFP will be computed with the addition of a GFP channel, with the reference electrode included, no field z-scores, all channels are included, and with the results written to an output file.

**INSERTSTIMEVENT** - Insert Stimulus Event.  This command allows you to insert a stimulus event at a specified point in a CNT file. To insert "Accept" and "Reject" type of events, use the REJECTBLOCK command.

1	int	Offset (number of points into the data file)
2	int	Stim code
3	int	Response code
4	double	Response latency
5	defined value	Accuracy (NORESPONSE, INCORRECT, CORRECT)

*Example.* INSERTSTIMEVENT 2500 10 0 400 corr. This enters a stimulus type code of 10 at the 2,500th data point (with the indicated response latency and accuracy).

The script may appear similar to the following, in which the event is inserted, and an event file is created:

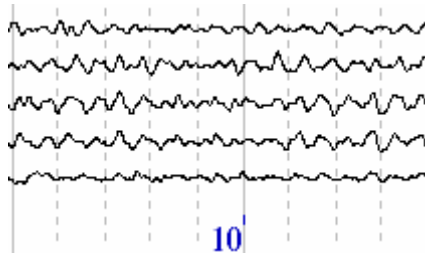
```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\closed.cnt"
```

```

INSERTSTIMEVENT 2500 10 0 400 corr
SAVEAS "$path\\newclosed.cnt"
OPENFILE "$path\\newclosed.cnt"
SAVEEVENT Y N N N N N "$path\\newclosed.ev2"
CLOSEALL

```

The event is added to the CNT file.



The event is seen in the created EV2 file.

```

1      10  0      1      400.0000 2500

```

**INSERTRESPONSEEVENT** - Insert Response Event. ⚡ This command allows you to insert a response event at a specified point in a CNT file. To insert "Accept" and "Reject" type of events, use the **REJECTBLOCK** command.

1	int	Offset (number of points into the data file)
2	int	Keypad code

*Example.* **INSERTRESPONSEEVENT 2500 4.** This enters a stimulus type code of 4 at the 2,500th data point.

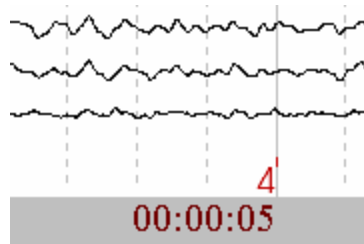
The script may appear similar to the following, in which the event is inserted, and an event file is created:

```

set path "c:\\ScanData\\Demo\\EEGs"
OPENFILE "$path\\closed.cnt"
INSERTRESPONSEEVENT 2500 4
SAVEAS "$path\\newclosed.cnt"
OPENFILE "$path\\newclosed.cnt"
SAVEEVENT N Y N N N N "$path\\newclosed.ev2"
CLOSEALL


```

The event is added to the CNT file.




The event is seen in the created EV2 file.

```
1    0 4 -1 0.0000 2500
```

**INTEGRATEWAVEFORM** - Integrate Waveform.  This command performs an accumulating sum across each sweep for each channel (refer to the EDIT manual for more details). Use it with AVG and EEG files (time domain only).


1	Boolean	Rectify
2	string	Output file (required for AVG and EEG files) (or "")

*Example.* INTEGRATEWAVEFORM N "\$path\\output file.avg". The integration is performed without rectifying the waveforms, and the output file is saved.

**LDR** - Linear Derivation.  The Linear Derivation transform enables the creation of new channels as arbitrary linear combinations of existing channels (see the EDIT manual for more details). It may be applied to CNT, EEG or AVG files. (If you apply an LDR file, additional display pages you may have created will not be available).

1	string	LDR file name (or "")
2	string	Output file name (or "")
3 (opt)	string	Units (defaults units if not specified)

*Example.* LDR "c:\\ScanData\\Batch Examples\\bipolar.ldr"  
"c:\\ScanData\\Batch Examples\\newfile.avg". A transformed data file is created after applying the LDR file. In this example, the default units, that is, whatever voltage units - uV, mV or V - that the EDIT program detects, are used. Enter a different y-axis label in the string if that is appropriate for your data.

**LOAD3DD** - Load 3DD File.  This command is used to load a 3DD electrode position file (from 3DSpaceDx) to whichever data file has the focus.

1	string	3DD file name (or "")
---	--------	-----------------------

*Example.* LOAD3DD {c:\\ScanData\\position.3dd}. The indicated 3DD file is loaded to a previously retrieved data file.

**MERGEVET** - Merge Event File. ☼ This command is used to merge behavioral data (the DAT file from STIM) with its matching continuous data file. The command is used only with CNT files. The MERGEVET command replaces the discontinued **TASKFILE** command.

1                      string                      Task file name (or "")

*Example.* MERGEVET "c:\\ScanData\\Batch Examples\\stimfile.dat". The full BATCH sequence might appear as follows (with the "path" variable substitution):

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\viscpt.cnt"
MERGEVET "$path\\viscpt.dat"
SAVEAS "$path\\newviscpt.cnt"
```

**Open/Save Commands.** In most instances, you should now use the OPENFILE and SAVEAS commands. AVGFILE, EEGFILE, CNTFILE, GETAVG, GETCNT, and GETEEG were the DOS commands for opening AVG, EEG and CNT files. These have been discontinued, and the OPENFILE command should be used instead.

**OPENFILE** - Open File. ☼☼☼☼☼ This command is used to open all types of data files in the EDIT program. As a general rule, you should use OPENFILE after every file you create. For example, if you output an ocular artifact corrected file, use OPENFILE to retrieve it before applying any other commands (otherwise you will likely be working with the uncorrected file!). *Note that you may not overwrite an open data file, and that you may not have more than one copy of the same data file open at the same time.* See also CLOSEALL, CLOSEFILE and SELECTFILE.

1                      string                      File name (or "")


*Example.* OPENFILE "c:\\ScanData\\data\\myfile.eeg". The indicated file will be opened.

**SAVEAS** - Save File As. ☼☼☼☼☼ The SAVEAS command is used to save data files. You cannot overwrite an EEG or CNT file that is currently open (use a different name).

1                      string                      Output file name (or "")

*Example.* SAVEAS "c:\\ScanData\\Batch Examples\\newfile.cnt". The indicated file will be saved to the designated folder.

To save a CNT file to the European Data Format (EDF), use the \*.rec extension instead of the \*.cnt extension.

**Peak Detection.**  The Peak Detection routine may be used to detect automatically the peak voltage within a user specified range in EEG or AVG files. With frequency domain EEG and AVG files, you will not see the peak markers unless you use the "line" style, rather than the "histogram" style of display (use **SETSPECTDISPLAY-STYLE** to select the line style). An ASCII file is produced that contains the latencies and voltages for the peaks at each channel. The principle command used in automatic peak detection is the **PEAKDETECTION** command. Related commands are the **EXCLUDEFORPEAKDETECTION**, **RESETFORPEAKDETECTION**, and **SAVEPEAK** commands. These are described below (in alphabetical order). An example combining the current commands is presented at the end of this subsection.

**EXCLUDEFORPEAKDETECTION** - Exclude Channel(s) for Peak Detection. Use this command to exclude channels from Peak Detection.

1...n                      list                      Electrode name(s)

*Example.* EXCLUDEFORPEAKDETECTION {Fp1 Fp2 F7 F8}.

**PEAKDETECTION** - Peak Detection. This command performs an automatic peak detection operation, placing markers at the detected locations. It may be applied to AVG or EEG files. To place multiple markers, just use multiple **PEAKDETECTION** commands.

1	string	Marker label
2	Boolean	User defined interval
3	int	Start search point (ignored if param 2 is False)
4	int	Stop search point (ignored if param 2 if False)
5	def value	Search type (NONE, MAX, MIN)
6	Boolean	Locked
7	string	Master channel label (ignored if param 6 is False)
8	Boolean	Exclude bad channels
9	Boolean	Exclude skipped channels
10(opt)	def value	Label type (USER, LATENCY, AMPLITUDE, BOTH)

*Example.* PEAKDETECTION "P300 Peak" Y 200 400 MAX Y "Pz" Y Y BOTH. The "P300 Peak" is detected between 200 and 400ms, based on the maximum voltage at the Pz channel. The 10th parameter is optional (USER is default).

**PEAKDETECTION\_EX** - Include Channel(s) for Peak Detection. This command differs from the **PEAKDETECTION** command in that it allows you to list the electrodes that you want to include for the termination channels (or "All"). See the "**\_EX Commands**" section above. The "Auto-write" feature will create

an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	string	Marker Label
2	Boolean	User Defined interval
3	float	Start search latency (ignored if param 2 if false)
4	float	Stop search latency (ignored if param 2 if false)
5	defined value	Search Type (NONE, MAX, MIN)
6	Boolean	Locked
7	string	Master Channel Label (ignored if param 6 is false)
8	Boolean	Exclude Bad channels
9	Boolean	Exclude Skipped channels
10	defined value	Label type (USER, LATENCY, AMPLITUDE, BOTH)
11	defined value	Color (RED, ORANGE, YELLOW, GREEN, BLUE, LTBLUE, BLACK, WHITE)
12	list	Channels to mark (or "All")

*Example.* PEAKDETECTION\_EX P100 Y 80 120 MAX N x Y  
Y USER GREEN {O1 O2 OZ PO1 PO2}. Peak detection is carried out on the selected channels only.

**RESETFORPEAKDETECTION** - Reset Channel(s) for Peak Detection. This command will reset all channels for peak detection. It is typically used where only some channels had been set for peak detection. It avoids having to reset each channel individually.

*Example.* The RESETFORPEAKDETECTION command is written on a line by itself, and uses no parameters.

**SAVEPEAK** - Save Peak File. The SAVEPEAK command is used to create/open an ASCII file (DAT extension), and to save the peak detection information to it. If you retrieve an existing DAT file, the new peak information will be added to the end of the file. If the -SubstituteBad option is selected, then the next parameter must be a string that will be placed in the location of peak data for all bad electrodes. This is a convenient way for flagging in subsequent analyses that these electrodes are bad.

1	string	File name (or "")
1(optional)	defined value	Option (-SubstituteBad)
2	string	Substitution string

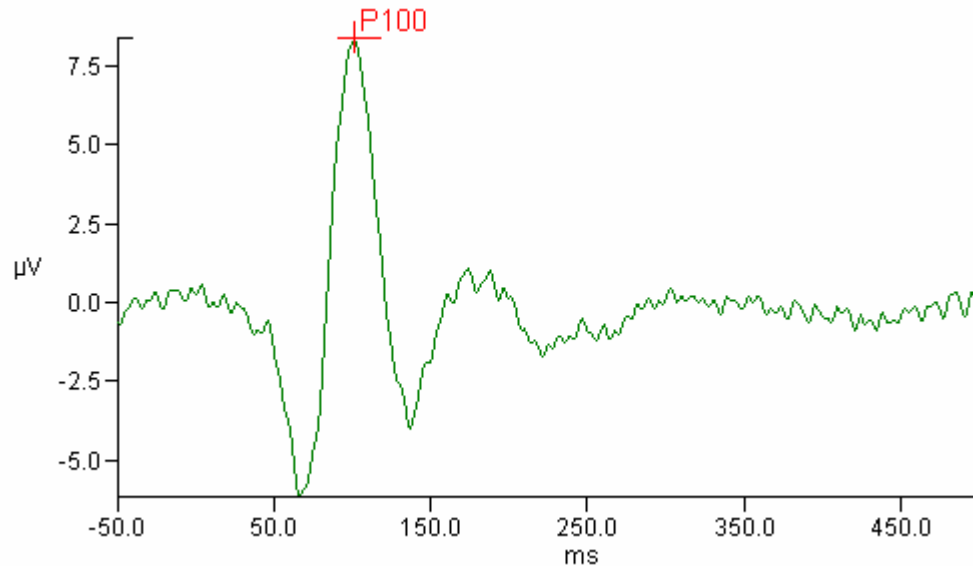
*Example.* SAVEPEAK "c:\\ScanData\\Batch  
Examples\\peakdata.dat" -SubstituteBad "Bad Electrodes". Below is a sequence of commands that includes saving the DAT file.

```

set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\vep.avg"
EXCLUDEFORPEAKDETECTION {VEOG HEOG}
PEAKDETECTION P 100 Y 80 120 MAX Y "OZ" Y Y BOTH
SAVEPEAK "$path\\peakdata.dat" -Sub "Bad Electrodes"
RESETFORPEAKDETECTION

```

The labelled peak marker is placed in all channels:



The DAT file that is created contains the latency and amplitude for the peak at each electrode and for each sweep (portion of the DAT file shown below):


Sweep	Number	Channel	Marker	Latency	Amplitude
1	FP1	P100	0.101000	-1.868643	
1	FP2	P100	0.101000	-2.033156	
1	F3	P100	0.101000	-2.007228	
1	F4	P100	0.101000	-2.950348	
1	C3	P100	0.101000	-0.623584	
1	C4	P100	0.101000	-2.942086	
1	P3	P100	0.101000	2.130232	
1	P4	P100	0.101000	1.385077	
1	O1	P100	0.101000	5.798689	
1	O2	P100	0.101000	6.215950	

*Note: SCAN 3.0 used a different format for the peak detection DAT files. There is an option in SCAN 4.3 to use the same format. Please see the Peak Detection section of the EDIT manual for instructions.*

**PRINT** - Print. This command will print the contents of the currently active window.


*Example.* PRINT. The content of the active window will be printed. See also SELECTFILE to change the focus to the window to be printed.



**RECTIFY** - Rectify File.  The RECTIFY transform is a simple “absolute value” operation: all positive waveform values are left unchanged, and all negative waveform values are inverted to their corresponding positive values. It may be used with EEG and AVG file types. The output file is required for EEG files and is omitted for AVG files.


1(optional)	string	Output file name (or "")
-------------	--------	--------------------------

*Example.* RECTIFY "c:\\ScanData\\Batch Examples\\rectfile.eeg". The file in working memory is rectified and the results are written to a new file.

**RECTIFY\_EX** - Rectify File.  The RECTIFY\_EX command differs from the RECTIFY command only in that you may select the channels to include (new parameter 1).

1	list	List of channels to include (or "ALL")
2(optional)	string	Output file name (or "")

*Example.* RECTIFY\_EX "ALL" "c:\\ScanData\\Batch Examples\\rectfile.eeg". All channels of the file in working memory are rectified, and the results are written to a new file.


**REFER** - Create New Reference.  The REFER command will compute a new reference using the electrodes designated as reference sites. (Note: this command exists in BATCH only. In point and click mode, use the GFP/Reference transform, and select the desired channels to be included in the new reference). The flags for Bad and Skip channels is not recognized in this command. All listed channels will be included, regardless of Bad or Skip designations.

1	Boolean	Compute GFP
2	Boolean	Include reference electrode
3	Boolean	Compute field z-scores
4	list	Channels to include (or "All")
5	string	Output file

*Example.* REFER N N N {A1 A2 FPZ} "\$path\\refer output file.eeg". A new EEG file is created using the designated reference.

The second parameter allows you to *Include the Reference Electrode* (when enabled), or exclude it (when disabled). The Reference Electrode is the electrode to which the relative electric potentials at the EEG leads are compared. When computing GFP, the program will use either the active EEG leads (N) alone, or the active EEG leads plus the presumed inactive reference electrode (N+1). If you enable the field, the computations will be performed with N+1, and the activity from the reference channel will be included. Otherwise, the total N will be used.

The *Compute Field Z-scores* option normalizes the GFP data by using the standard deviation of the topography/field, at each time point. Think of these as spatial Z scores. This is quite useful when making a movie. Essentially, the data are "autoscaled" at each time point, so the map color scale is set in terms of standard deviations.

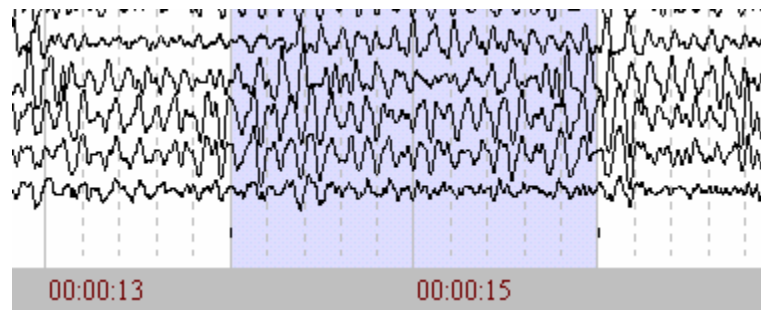
**REJECTBLOCK** - Reject Block.  Use this command to reject blocks of data in CNT files. (You will need the AD rate to compute the offset:  $\text{offset} = \text{AD}/1000 \times \text{ms}$ ).


1	int	Start offset
2	int	Stop offset

*Example.* REJECTBLOCK 7000 8000. The section of the CNT file encompassing the 7000-8000 data points will be marked as a rejected block. The more complete script might appear as follows:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\closed.cnt"
REJECTBLOCK 7000 8000
```

The section between the 14-16 seconds is rejected (AD = 500).



**RESPWIN** - Response Window.  The RespWin (Response Window) program performs an offline analysis of behavioral data in an event (.EVT) file that consists of a stream of stimulus and response events. The user defines a response window and one or more paired subsets of stimuli that correspond to target events and associated distractor events. A hierarchy of rules is applied to resolve ambiguities in forming associations between stimuli and responses. Hits, correct rejections, false alarms, and misses are tallied, and a nonparametric signal detection analysis is given. Reaction times are computed for each stimulus-response association (see the EDIT manual for details).

1	double	Window minimum (ms)
2	double	Window maximum (ms)
3	Boolean	No Intervening Stimuli
4	string	Output event file name (.ev2 extension)
5	string	Output summary file name (.sum extension)
6	Boolean	Include header in the event file

x	string	Descriptor
x	string	Target vector (1, 2, 3-6, etc.)
x	int	Target correct response
x	string	Distractor vector (1, 2, 3-6, etc.)
x	int	Distractor correct response (or 0)

*Example.* RESPWIN 100 1000 N "\$path\\event file.ev2"  
"\$path\\summary file.sum" Y "Target" "1" 1 "2-10" 0. In this example, responses occurring 100-1000ms after the stimuli are potentially valid. Response type "1"s are associated with stimulus type "1"s. Stimuli with types of 2-10 have no responses. The output files are specified. If you have multiple target/distractor pairs, repeat the 7 through 11 parameters for each pair (all within the same RESPWIN command).

**SAVEEVENT** - Save Event File. 🌀 The SAVEEVENT command performs a SCAN of a continuous (CNT) file and writes an event file (EVT or EV2) to disk containing the file position and tag of each event. If a task file is loaded prior to the SAVEEVENT command, behavioral data (accuracy, reaction time, responses and stimuli) will be included in the event file. In versions prior to 3.0 the event file can be recalled prior to the EPOCH command to increase the speed of flexibility of event windowing. This is not necessary in 3.0 versions and later since the event information is stored in a table for rapid lookup during epoch creation.

1	Boolean	Include Stimulus events
2	Boolean	Include KeyPad events
3	Boolean	Include Keyboard events
4	Boolean	Include Rejected sweeps
5	Boolean	Use seconds
6	Boolean	Include header
7	string	Output event file name (*.evt= old style, *.ev2= new style; or "")


*Example.* SAVEEVENT Y N N N N Y "\*.ev2". An event file (ev2 type, using points instead of byte offsets) is created using only the stimulus events.

**SAVEEVENT\_EX** - Save Event File, Extended. 🌀 The SAVEEVENT\_EX command adds the additional options to save the DC Correction and Start/Stop events to the event file.

1	Boolean	Include Stimulus events
2	Boolean	Include KeyPad events
3	Boolean	Include Keyboard events
4	Boolean	Include Rejected sweeps
5	Boolean	Use seconds
6	Boolean	Include header
7	Boolean	Include DC corrections

8	Boolean	Include Start/Stop events
9	string	Output event file name (*.evt= old style, *.ev2= new style; or "")


*Example.* SAVEEVENT Y N N N N Y N N "\*.ev2". An event file (ev2 type, using points instead of byte offsets) is created using only the stimulus events.

**SELECTFILE** - Select Open Data File.  This command allows you to switch the current working file to one of several data files that you may have open. Once selected, you may perform further processing of the selected file, while leaving the other files open. The BATCH processor directs all data manipulation commands to the current working file. The current working file will remain in effect until another file is opened, another file is selected, or the current file is closed. See also **OPENFILE**, **CLOSEFILE**, and **CLOSEALL**.

1	string	Data file to be selected (" " will fail)
---	--------	--

*Example.* SELECTFILE "file1.avg". The "focus" is switched to the indicated file. Note that you do NOT use the path with this command. A script using the command is displayed below.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\file1.avg"
OPENFILE "$path\\file2.avg"
OPENFILE "$path\\file3.avg"
ARRANGEWINDOWS TILEVERT
SELECTFILE "file1.avg"
CUTEPOCH 0 200 "$path\\cutfile1.avg"
CLOSEALL
OPENFILE "$path\\cutfile1.avg"
ZOOMIN "PZ"
```

**Smoothing Data.**  Averaged data files may be smoothed, that is, a given data point may be averaged with adjacent points to create a smoother waveform. The principle command used is **SMOOTH**, and related commands are **EXCLUDEFORSMOOTH** and **RESETFORSMOOTH**. These are describe below.

**EXCLUDEFORSMOOTH** - Exclude Channel(s) for Smoothing. Use this command to exclude channels from Smoothing.

1...n	list	Electrode name(s)
-------	------	-------------------

*Example.* EXCLUDEFORSMOOTH {Fp1 Fp2 F7 F8}.

**RESETFORSMOOTH** - Reset Channel(s) for Smoothing. This command will reset all channels for smoothing (so that none are excluded). It is typically used where only some channels had been set for smoothing. It avoids having to reset each channel individually.

*Example.* The RESETFORSMOOTH command is written on a line by itself. No parameters are used.

**SMOOTH** - Smooth. The SMOOTH command is used to average each data point with adjacent points to create a smoother waveform. The command may be used with EEG or AVG files. The smoothed results are written to a EEG new file. Smoothing with AVG files is done "in place" - you will need to save the smoothed AVG file with the SAVEAS command, if desired.


1	int	Number points in smoothing (must be an odd number, at least 3)
2	int	Number of passes through the data
3(optional)	string	EEG output file name (or "")

*Example.* SMOOTH 3 3 "c:\\ScanData\\Batch Examples\\smoothed data.eeg". Three adjacent data points are smoothed, and three passes are made.

**SMOOTH\_EX** - Include Channels for Smooth. This command differs from the SMOOTH command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All"). The smoothed results are written to a EEG new file. Smoothing with AVG files is done "in place" - you will need to save the smoothed AVG file with the SAVEAS command, if desired.

1	int	Number of points in smooth (must be an odd number, at least 3)
2	int	Number of passes through data
3	list	List of electrodes to smooth (or "All")
4(optional)	string	EEG output file name

*Example.* SMOOTH\_EX 3 3 {C3 CZ C4} "c:\\ScanData\\demo\\output file.eeg". The indicated electrodes will be smoothed.

**SNR** - **NEW** Signal to Noise Ratio.  (time domain only) The Signal to Noise Ratio (SNR) transform computes "noise" over a specified interval, "signal" over a specified interval, and then the ratio of signal over noise. "Signal" and "noise" are defined as the variance among data points in the selected interval. The SNR then is the ratio of the variance (signal) over the variance (noise). You can specify the interval to use for the

noise and signal. Typically, you would use the pre-stimulus interval to estimate the noise, and some or all of the post-stimulus interval to estimate the signal. If there is no prestimulus interval, the 20th percentile is used for the Noise estimate.

1	defined value	Noise interval (PRESTIMINTERVAL, PERCENTILE, USERDEFINED)
2	double	Interval Noise start latency
3	double	Interval Noise stop latency
4	defined value	Signal interval (POSTSTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
5	double	Interval Signal start latency
6	double	Interval Signal stop latency

*Example.* SNR USERDEFINED -500 0 USERDEFINED 0 300. The noise interval is defined as -500 to 0ms, and the signal interval is defined as 0 to 300ms. To view the SNR values later, save the file with the modifications.

**Sort** - Sort Sweeps. \*\*\* The Sort command will create a new EEG file that contains only the sweeps that you designate (used with EEG files only). For latency and accuracy information, you will need to have merged the DAT file from STIM (MERGEEVT) with the CNT file prior to epoching (EPOCH) the CNT file. The SORT command is used to apply the sorting criteria you create using the CREATSORT command. In the example below, the "sortfile" was already created. The results are written to a new file.

1	string	Sort name
2	string	Output file name (or "")

*Example.* SORT "sortfile" "c:\\ScanData\\Batch Examples\\sorted sweeps.eeg". The "sortfile" criteria will be applied, and the output file will be saved. The complete sequence might appear similar to the following script.

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\p300.eeg"
CREATESORT sortfile
sortfile -TrialEnabled YES
sortfile -TrialCriteria "1-50"
sortfile -TypeEnabled YES
sortfile -TypeCriteria "1"

SORT "sortfile" "$path\\sorted sweeps.eeg"
```

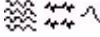
In this example, only the sweeps with stimulus type codes of 1, in the first 50 sweeps, will be written to a new EEG file.

**SPATIALFILTER** - Spatial Filter. ~~~\*\*\*~ The spatial filter transform removes and/or retains signals in spatial subspaces of the full measurement space in the context of

a control signal (refer to the EDIT manual for more details). It may be used with CNT, EEG or AVG files. To use the command, you will need to have already created the Removal, Retention, and/or Control LDR files (see the VEOG correction examples in the Spatial Filter section of the EDIT manual for ways to create these). Different reconstitution options are available depending upon which type(s) of LDR files you are using (Removal, retention, etc.). If you are not completely familiar with using the Spatial Filter, you should set up the procedure in point&click mode first, then transfer the settings manually to the Tcl command line.

1	string	Removal LDR
2	string	Retention LDR
3	string	Control LDR
4	string	Output spatial filter LDR file
*****Reconstitution of Original Channels*****		
5	Boolean	Reconstitute original channels
6	Boolean	Remove unwanted signals
7	Boolean	SNR Transform
8	Boolean	Project signals of interest only
9	Boolean	Include skipped channels
*****Derivation of Signal channels*****		
10	Boolean	Derive channels for unwanted signals
11	Boolean	Derive channels for Signals of interest
12	string	Output file name (CNT files only)

*Example.* SPATIALFILTER "artifact.ldr" "" "cleaneeeg.ldr" "LDRoutput.ldr" N Y N O N Y N "". In this example, we are using a Removal LDR and a Control LDR file. The created LDR file is saved. Unwanted signals are removed, and Skipped channels are excluded.

**SPATIALSVD** - Spatial SVD.  The spatial singular value decomposition (SVD) transform generates spatial component topographies, fraction of total variance explained by each component, and a linear derivation file for deriving component time series. This transform can be applied to any time domain file type (AVG, EEG, or CNT). Please refer to the EDIT manual for more details. See also **CREATESORT**.

1	defined value	Covariance/Cross product (COVARIANCE, CROSSPRODUCT)
2	Boolean	Include skipped
3	double	Start latency
4	double	Stop latency
5	double	Percent retained variance (not recognized if DATAMATRIX is selected)
6	string	Output LDR file
7	defined value	Output LDR type (COMPONENTS, DATAMATRIX)
8	string	Sort name



*Example.* SPATIALSVD COV NO 250 400 95 "\$path\\spatSVD.LDR"  
COMP "sortfile". The SVD will be calculated according to the parameters specified. A more complete sequence, using a sort, is shown below.

```
set path "c:\\ScanData\\Batch Examples"  
OPENFILE "$path\\p300.eeg"  
CREATESORT sortfile  
sortfile -TrialEnabled YES  
sortfile -TrialCriteria "1-50"  
sortfile -TypeEnabled YES  
sortfile -TypeCriteria "1"  
SPATIALSVD COV NO 250 400 95 "$path\\spatSVD.LDR"  
COMP "sortfile"
```

**Spectral analyses.** Average waveforms consisting of a power-of-2 number of points (e.g., 64, 128, 256, 512, 1024, 2048, etc.) can be Fourier analyzed to obtain an amplitude (i.e., root power) spectrum at each electrode. If your average waveform consists of a non-power-of-two number of points, use the Spline transform first (see below). The SPECTRUM command is used with AVG files. It is also possible to perform FFT analyses on EEG files (assuming the number of points per epoch is a power of 2). The FORWARDFFT command accomplishes this function, and the INVERSEFFT command restores the FORWARDFFT data back to its original form.

Prior DOS BATCH files used the SETSPECT command to set parameters in the spectral analyses. This is now accomplished in the SPECTRUM command, and the SETSPECT command is no longer recognized.

**FORWARDFFT/INVERSEFFT** - Forward FFT/Inverse FFT. <sup>\*\*\*</sup> The FORWARDFFT command performs an FFT on epoched files. The number of points in each channel and sweep must be a power of 2 (256, 512, etc.). Use the INVERSEFFT command to restore the data.

1                      string                      Output file name (or "")

*Example.* FORWARDFFT "c:\\ScanData\\Batch  
examples\\forwardFFT.eeg".  
INVERSEFFT "c:\\ScanData\\Batch Examples\\originaldata.eeg".  
Below is an example script using FORWARDFFT and INVERSEFFT  
using the SPLINE command.

```
set path "c:\\ScanData\\Batch Examples"  
OPENFILE "$path\\p300.eeg"  
  
# GENERATES A SPLINE INTERPOLATED DATA SET CONSISTING OF 512 POINTS  
PER SWEEP  
SPLINEFIT 512 "$path\\splinep300.eeg"
```



```

# OPENS SPLINE INTERPOLATED FILE FOR FFT
OPENFILE "$path\splinep300.eeg"

# PERFORMS FFT AND WRITES OUTPUT FILE
FORWARDFFT "$path\FWp300.eeg"

# CLOSES ALL FILES - A WINDOW CLEANUP
CLOSEALL

# OPENS THE FFT-TRANSFORMED EEG FILE FOR INVERSE FFT OPERATION
OPENFILE "$path\FWp300.eeg"

# PERFORMS INVERSE FFT AND WRITES OUTPUT FILE
INVERSEFFT "$path\INVp300.eeg"

# OPENS INVERSE FFT FILE FOR INSPECTION
OPENFILE "$path\INVp300.eeg"

# PAUSES EXECUTION OF SCRIPT - CONTINUE BY PRESSING RESUME BUTTON
ON BATCH WINDOW
PAUSE



# CLOSES ALL OPEN DATA WINDOWS
CLOSEALL

```

**SPECTRUM** - Spectrum.  The SPECTRUM command will compute a Fast Fourier Transform on an AVG file.

1	defined value	Spectrum type (MAGNITUDE, PHASE)
2	defined value	Scaling method (ignored if param 1 is PHASE) (AMPLITUDE, POWER)
3	defined value	Window type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
4	double	Window taper percent
5	string	Output file name (or "")

*Example.* SPECTRUM MAG AMP COS 10 "c:\ScanData\Batch Examples\outputFFT.avg". This command will perform an FFT on the file in working memory, calculating the Amplitude, windowing with a 10% cosine taper, and writing the output file.

**SPLINEFIT** - Spline Fit.   The SPLINEFIT command will perform a spline interpolation of the current waveform average. The command can be used to fit an arbitrary number of points to the waveform, but it is particularly useful when large numbers of waveform averages have been created with a non-power-of-two number of points, but the user desires to compute FFT spectra on these waveforms. When epoched or continuous data is acquired under a set of parameters that leads to waveform averages

with a non-power-of-two number of points, the number of points in the waveform average can be changed with the **SPLINEFIT** command. The **SPLINEFIT** command can also be used to change the number of points in one set of waveform averages so that they can be group averaged with another set of waveforms which were created with a different number of points. The command is used with EEG and AVG files.

1	int	Points (must be a power of 2)
2	string	Output file name (or "")

*Example.* **SPLINEFIT 1024 {c:\ScanData\newfile.eeg}**. The file in working memory will be splinefitted to 1024 points per second.

**Statistical Comparisons.** ~ Statistical comparisons between files/groups include correlations, t-tests, and z-scores. The DOS BATCH files used **GETCOMP** and **SAVECOMP** commands to retrieve the comparison file and save the results. **SAVECOMP** is obsolete with the Tcl commands. The **GETCOMP** command is still used for superimposing a comparison file on top of a reference file (see Miscellaneous commands above).

#### Correlational measures.

**CROSSCORR** - Cross Correlation. ~ The Cross Correlation coefficient statistic is created by computing the correlation between electrodes across a lag series. It can be used to examine the relationships between or among electrodes by shifting the waveforms in time, and then recomputing the correlations (see the EDIT manual for more details). Retrieve one file, and then use the **CROSSCORR** command.

1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval stop
4	string	Output data file name (or "")

*Example.* **CROSSCORR "\$path\comparison file.avg" 0 200 "\$path\output file.avg"**. The cross-correlation is computed for the 0-200ms interval between files, and the output file is created.

**CROSSCORR\_EX** - Cross Correlation, extended. ~ The **CROSSCORR\_EX** command is the same as the **CROSSCORR** command, except for the new 5th parameter that lets you specify the channels to be included.

1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval stop

4	string	Output data file name (or "")
5	list	Channels to compute

*Example.* CROSSCORR "\$path\\comparison file.avg" 0 200 "\$path\\output file.avg" {FZ CZ PZ OZ}. The cross-correlation is computed for the 0-200ms interval between files for midline electrodes only, and the output file is created.

**INTRACLASSCORR** - Intra-class Correlation. ~ The intra-class correlation statistic is a measure of overlap and related variability between two waveforms (from two AVG files). Retrieve one file, then enter the BATCH command on the next line.

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.dat) (or "")

*Example.* INTRACLASSCORR "c:\\ScanData\\Batch Examples\\reffile.avg" 100 200 "c:\\ScanData\\Batch Examples\\corrfile.dat". The intra-class correlation is computed between 100-200ms, and the ASCII output file is created.

**INTRACLASSCORR\_EX** - Intra-class Correlation, extended. ~ This command differs from the INTRACLASSCORR command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.dat) (or "")
5	Boolean	Display results
6	list	Channels to include

*Example.* INTRACLASSCORR\_EX "c:\\ScanData\\Batch Examples\\reffile.avg" 100 200 "c:\\ScanData\\Batch Examples\\corrfile.dat" Y {O1 OZ O2 P3 P4}. The intra-class correlation is computed for the indicated channels between 100-200ms, the ASCII output file is created, and the results are displayed.

**PEARSONS** - Pearson's r Correlation. ~ This operation computes Pearson's r correlation coefficients between paired electrodes within a

specified latency range. This option is available for time and frequency domain AVG files. Retrieve the first file, then follow the format below.

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.DAT) (or "")

*Example.* PEARSONS "c:\\ScanData\\Batch Examples\\referfile.avg" 0 500 "c:\\ScanData\\Batch Examples\\referout.dat". Retrieve one data file first (using, for example, OPENFILE), then retrieve the Reference file. Pearson r correlations are computed during the 0-500ms range, and the results are written to an ASCII file.

**PEARSONS\_EX** - Pearson's r Correlation, extended. ~ The PEARSONS\_EX command is the same as the PEARSONS command except for the new 5th parameter that lets you specify the channels to include.

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.DAT) (or "")
5	Boolean	Show results
6	list	List of channels to include

*Example.* PEARSONS "c:\\ScanData\\Batch Examples\\referfile.avg" 0 500 "c:\\ScanData\\Batch Examples\\referout.dat" Y {T3 C3 CZ C4 T4}. Retrieve one data file first (using, for example, OPENFILE), then retrieve the Reference file. Pearson r correlations are computed during the 0-500ms range, for the specified channels, and the results are written to an ASCII file as well as displayed.

## T-test comparisons.

**PAIRED** - Paired t-test. ~ A paired t-test is used when comparing two related or matched groups, or when comparing test-retest measures on the same group. See the EDIT manual for more details. The following summary steps should be taken to perform a paired t-test in BATCH (these steps could all be performed in BATCH):

1. For each individual in both groups, make and save a waveform subtraction between the two conditions of the experiment. For example, in a RARE and FREQ comparison, retrieve the first

subject's RARE waveform, select Subtract from the Transforms list, and then select the same subject's FREQ data file. Repeat for all pairs of files.

2. Make a group average difference waveform composed of the difference waveforms obtained in Step 1.

3. If the group average difference waveform has already been created, retrieve it as the working file.

4. Execute the PAIRED command with the output file specified.

1	string	Output file name
---	--------	------------------

*Example.* PAIRED "c:\\ScanData\\Batch Examples\\output t-scores.avg". In the following example, we have already created the difference file (diff file.avg).

```
OPENFILE "c:\\ScanData\\Batch Examples\\diff file.avg"
PAIRED "c:\\ScanData\\Batch Examples\\output t-scores.avg"
```

**TSCORE** - t-test. ^v The TSCORE command calculates t-test comparisons between two waveform files which have the optional variance waveforms as well as the normally displayed mean-value waveforms. For example, this option would typically be used for the comparison data from two independent groups. Retrieve one group averaged file, then use the TSCORE command.

1	string	Reference file name (or "")
2	string	Output file name (or "")

*Example.* TSCORE "c:\\ScanData\\Batch Examples\\compfile.avg" "c:\\ScanData\\Batch Examples\\resultsfile.avg".


**ZSCORE** - Z-score. ^v The ZSCORE command calculates Z-scores between two waveform files. This command would typically be used for statistical comparison of a waveform from an individual with an averaged waveform from a group.

1	string	Reference file name (or "")
2	string	Output file name (or "")

*Example.* ZSCORE "c:\\ScanData\\Batch Examples\\groupavg.avg" "c:\\ScanData\\Batch Examples\\zscores.avg". The more complete script sequence might appear as:

```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\individual.avg"
ZSCORE "$path\\group.avg" "$path\\zscores.avg"
OPENFILE "$path\\zscores.avg"
```



The individual subject file is opened, then the reference group is called, the zscores are calculated and saved, and the output file is retrieved.


**SUBTRACT** - Subtract Files.  The SUBTRACT command is used to perform a point by point mathematical subtraction of the selected AVG file to the one specified by the first parameter and create a new output file. This is used, for example, to compute the difference waveform between the rare and common responses of a p300 recording. The command is also used to subtract an AVG file from each sweep of an EEG file.

1	string	File name to subtract (or "")
2	string	Output file name (or "")

*Example.* SUBTRACT "c:\\ScanData\\Batch Examples\\compfile.avg" "c:\\ScanData\\Batch Examples\\resultsfile.avg". Here is a simple example (be sure to use the OPENFILE command to retrieve the output file):


```
set path "c:\\ScanData\\Batch Examples"
OPENFILE "$path\\sepnoblk.avg"
SUBTRACT "$path\\sepnoblk.avg" "$path\\difference.avg"
OPENFILE "$path\\difference.avg"
ZOOMIN "FZ"
```

**UPDATECHANGES** - Update Changes to Data file.  The UPDATECHANGES command is used to save changes to the open data file that has the focus. It has the same function as the  icon on the Toolbar. The command will overwrite the open file, saving the changes with the file when you close it. The command exists on a line by itself, and uses no parameters. For example, say retrieve a file and perform peak detection in BATCH, and you wish to save the changes while leaving the file open. The UPDATECHANGES will update the open file, saving the changes when you close the file. (The command does not function with COH files).

**VOLTAGETHRESHOLD** - Voltage Threshold.  The Voltage Threshold option allows you to insert event markers into a CNT file (only) on the basis of a detected voltage in a specified channel, and to reject sections of a CNT where detected voltages exceed a specified threshold.

1	defined value	Operation type (INSERTEVENTS, REJECTSEGMENTS)
2	string	Trigger Channel
3	defined value	Threshold Type (GREATERTHAN, LESSTHAN, ABSOLUTEVALUE)
4	int	Stim code (ignored if operation is not Insert events)
5	double	Refractory period
6	double	Threshold

*Example.* VOLTAGETHRESHOLD INSERT "VEOG" GREAT 10 400 50. In this example, a trigger with a type code of 10 will be inserted at any point where the voltage in the VEOG channel exceeds 50uVs (with a 400ms refractory period, meaning that no triggers will be inserted for at least 400ms after a trigger has been placed, regardless of the amplitude).

**Windowing Data Files.**  The Window transform applies a window taper to single-sweep epochs (AVG or EEG files). Application of a window can be useful to minimize edge effects. WINDOWDATA is the principle command, and EXCLUDEFORWINDOWDATA and RESETFORWINDOWDATA are related commands.

**EXCLUDEFORWINDOWDATA** - Exclude Channel(s) for Windowing. Use this command to exclude channels from Windowing.

1...n	list	Electrode name(s)
-------	------	-------------------

*Example.* EXCLUDEFORWINDOWDATA {Fp1 Fp2 F7 F8}.

**RESETFORWINDOWDATA** - Reset Channel(s) for Windowing. This command will reset all channels for windowing. It is typically used where only some channels had been set for windowing. It avoids having to reset each channel individually.

*Example.* The RESETFORWINDOWDATA command is written on a line by itself, and uses no parameters.

**WINDOWDATA** - Window Data. The Window transform applies a window taper to single-sweep epochs (EEG or AVG files). Application of a window can be useful to minimize edge effects.


1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Window percent
3(opt)	string	Output file name (or "")

*Example.* WINDOWDATA HAN 10 {c:\ScanData\windowed data.eeg". A Hanning window with a 10% taper is applied, and the new file is saved.

**WINDOWDATA\_EX** - Include Channels for Window Data. This command differs from the WINDOWDATA command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Window percent
3	list	List of electrodes to include (or "All")
4(opt)	string	Output file name (or "")

*Example.* WINDOWDATA\_EX COS 10 {All}  
{c:\ScanData\windowed data.eeg". All of the channels are selected for windowing.

**WRITENUMSWEEPS** - Write Number of Sweeps.  This command will create an ASCII output file containing the number of sweeps for each channel (it is typically used with averaged files, where you wish to see/store how many sweeps were included in each channel's average).

1	string	Numsweeps file (or "")
---	--------	------------------------

*Example.* WRITENUMSWEEPS "c:\\ScanData\\Batch Examples\\numsweeps.dat". The output file can be read with any text editor software.

#### **Discontinued DOS Commands for EDIT.**

ACCURACY  
 AVGFILE  
 CARTESIAN/POLAR  
 CLOSEPEAK  
 CNTAVG  
 CNTFILE  
 CNTSAVE  
 EEGFILE  
 ELECTSCORE  
 EPOCH\_OVERLAP  
 EXPORTMAC  
 FCARTOON



**FSP\_STOPMETHOD**  
**FSP\_STOPMODE**  
**GETAVG**  
**GETCNT**  
**GETEEG**  
**GETMAP**  
**GETPEAKTEMP**  
**GROUP**  
**HOLD**  
**INDIVIDUAL**  
**LARGEMAP**  
**LATENCYRNG**  
**MAXSWEEPS**  
**MEANFREQ**  
**MINSWEEPS**  
**OPENPEAK**  
**PLAYMOV**  
**READEVENT**  
**RECORDMOV**  
**RESPCODE**  
**RESPONSERNG**  
**RETAIN**  
**REVERT**  
**SAVEAVG** - Use the **SAVEAS** command instead.  
**SAVECOMP**  
**SAVECSA** - Use the **SAVEAS** command instead.  
**SAVEDAT** - Use the **SAVEAS** command instead.  
**SEARCH**  
**SEQUENCE**  
**SETART** - (see **SETCHANATTRIBUTE**)  
**SETARTCORRASSIGN**  
**SETARTCORRPARAM DIRECTION**  
**SETARTCORRPARAM DURATION**  
**SETARTCORRPARAM MINIMUM**  
**SETARTCORRPARAM PERCENT**  
**SETARTCORREVIEW MAXIMA**  
**SETARTCORREVIEW SWEEPS**  
**SETARTCORREVIEW WEIGHTS**  
**SETBAD**  
**SETBASE** - (see **BASECOR**).  
**SETFILTRECTIFY** - This command is superceded by the **FILTER** command.  
**SETREF**  
**SETSKIP** - (see **SETCHANATTRIBUTE**)  
**SETSPECT** - This command has been superceded by the **AVERAGE**  
                   command.  
**TCARTOON**

TASKFILE  
 TCARTOON  
 TRIALRNG  
 TYPERNG  
 VARIANCE  
 VECTORSORT  
 WAVEFORM

## Advanced TCL Scripts

The following scripts illustrate more advanced Tcl applications.

**Complete BATCH Sequence.** This sample BATCH file is a bit more complex, illustrating a variety of transforms, and demonstrates loops and substitutions.

```
# THIS BATCH PROCESSES DATA FILES FROM CONTINUOUS FORMAT, TO
# EPOCH, AND FINALLY TO AVERAGE FILES. INCLUDED ARE A NUMBER OF
# PROCESSING STEPS INCLUDING OCULAR CORRECTION, BASELINE
# CORRECTION AND LINEAR DETRENDING.

# SPECIFIES PATH TO DATA FILES COLLECTED USING ODDBALL PARADIGM
set path "e:\\"

# SPECIFIES LIST OF FILES (LIST MUST BE ONE ITEM OR MORE)
set filelist {ERP1 ERP2}

# SPECIFIES EXTENSIONS FOR STANDARD AND DEVIANT AVERAGES
set ExtStan "_C"
set ExtDev "_D"

#STARTS PROCESSING LOOP
foreach element $filelist {

  #OPENS AND EPOCHS CONTINUOUS EEG FILE
  #SAVES A TEMP.EEG FILE FOR FURTHER PROCESSING
  set file $path$element
  OPENFILE "$file.cnt"
  EPOCH PORT 0 -100 400 N Y N N NULL "e:\Temp.eeg"

  #OPENS AND PERFORMS OCULAR CORRECTION ON EPOCHED FILES
  OPENFILE {e:\Temp.eeg}
  ARTCOR pos 10 20 400 VEOG LDR+ "$file.ldr" {e:\TempEyeCorr.eeg} Y
  Y

  #OPENS AND PERFORMS BASELINE CORRECTION ON OCULAR CORRECTED EPOCHED FILES
  #STORES LDR TRANSFORM TO ROOT NAME
  OPENFILE {e:\TempEyeCorr.eeg}
  BASECOR PRE 0 0 N N "e:\Temp_BCorr.eeg"
```

```

#CLOSES ALL OPEN FILES AND OPENS BASELINE CORRECTED EEG FILE FOR ARTIFACT
#REJECTION
#REJECTS EPOCHS WHERE PEAK AMPLITUDE EXCEEDS +/- 150 microVolts between 0 and 400 ms
#SAVES PROCESSED EEG FILE BACK TO ROOT NAME
CLOSEALL
OPENFILE {e:\Temp_BCorr.eeg}
SETART "ALL" ON ; SETART "HEOG" OFF; SETART "VEOG" OFF
ARTREJ CRIT N 0 400 N -150 150 Y Y
SAVEAS "$file.eeg"

#CLOSES ALL OPEN FILES AND DELETES TEMPORARY EEG FILEs
CLOSEALL
DELETE {{e:\Temp.eeg}}
DELETE {{e:\TempEyeCorr.eeg}}
DELETE {{e:\Temp_BCorr.eeg}}

#OPENS PROCESSED EEG FILE AND GENERATES AVERAGES BASED ON
#STIMULUS TYPE WITH ROOT NAME AND EXTENSION THAT SPECIFIES SORT TYPE

#AVERAGE RESPONSE TO DEVIANT STIMULUS
OPENFILE "$file.eeg"
CREATESORT DeviantSort;DeviantSort -TypeEnabled YES -TypeCriteria "10"
AVERAGE TIME 0 0 0 0 0 DeviantSort "e:\\Temp1.avg"
OPENFILE "e:\\Temp1.avg"
DETREND ENTIRE 0 0
BASECOR PRE 0 0 N N
GFP 0 0 0 "$file$ExtDev.avg"
CLOSEALL
DELETESORT DeviantSort

#AVERAGE RESPONSE TO STANDARD STIMULUS
OPENFILE "$file.eeg"
CREATESORT StandardSort; StandardSort -TypeEnabled YES -TypeCriteria "1-
9"
AVERAGE TIME 0 0 0 0 0 StandardSort "e:\\Temp2.avg"
OPENFILE "e:\\Temp2.avg"
DETREND ENTIRE 0 0
BASECOR PRE 0 0 N N
GFP 0 0 0 "$file$ExtStan.avg"
CLOSEALL
DELETESORT StandardSort

#CLOSES ALL OPEN FILES AND DELETES ALL TEMP.AVG FILES
DELETE {{e:\Temp1.avg}}
DELETE {{e:\Temp2.avg}}
}

```

**Advanced Artifact Rejection Application.** The following BATCH file demonstrates how to reject approximately 50% of the epochs in a file by repeatedly widening the

reject window until less than 50% of the epochs are rejected. The starting criteria (in this case 20) and the amount to open the window (in this case 10) could be modified for speed or greater precision based on the data being processed.

```
ENABLEOVERWRITEPROMPT N
set name [GETINPUT "name of file to reject" "file name"]
OPENFILE "c:\\ScanData\\Batch Examples\\$name.eeg"

set evcount [GETEPOCHCOUNT]
set numrej [GETEPOCHCOUNT]
set extname failed
set PercentRej 100.0
for {set mv 20} {$PercentRej > 50.0} {incr mv 10} {

    set pmv $mv
    set nmv -$mv

    SETART {ALL} Y
    ARTREJ CRITERIA Y X X Y $nmv $pmv Y N
    set numrej 0
    for {set epochnum 0} {$epochnum < $evcount} {incr epochnum} {
        if {[GETEPOCHINFO $epochnum -ACCEPT] == 0} {
            incr numrej
        }
    }
    set PercentRej [e xpr {100.0*$numrej/$evcount}]
    set extname $mv
}

INSTRUCT "percentage reject: $PercentRej"
SAVEAS "c:\\ScanData\\Batch Examples\\$name$extname.eeg"
OPENFILE "c:\\ScanData\\Batch Examples\\$name$extname.eeg"
```

**Changing Information in the Event Table.** Beginning with SCAN 4.3, you may access the event table in CNT files. In this example, we will demonstrate how to add a value of 10 to all stimulus type events. This involves "getting" existing event information from the event table, and "setting" new event information in the event table. Also in this example, note the use of the "pickoff" and format commands to build the output file path and name.

```
set path [GETINPUTFILE "select file" "cnt" "c:\\ScanData\\Batch Examples\\"]
if {$path == "Cancel"} {
    return
}
set valuetoadd [GETINPUT "Enter event value to add to each stimulus code"
"value"]
```

```

OPENFILE $path
set evcount [GETEVENTCOUNT]
for {set index 0} {$index < $evcount} {incr index} {
    set type [GETEVENTINFO $index -EventType]
    if {$type == Stimulus} {
        set stimcode [GETEVENTINFO $index -StimulusCode]
        SETEVENTINFO $index -StimulusCode [expr
{$stimcode+$valuetoadd}]
    }
}

set Directory [PICKOFFDIRECTORY $path]
set Filename [PICKOFFNAMEONLY $path]
set Ext [PICKOFFEXTENSION $path]

set OutPath [format "%s%s_Incremented_By_%d%s" $Directory $Filename
$valuetoadd $Ext]
SAVEAS $OutPath
CLOSEALL

```

The first line sets "path" as the variable name for the input file and path (as determined when you select the file using the **GETINPUTFILE** option). On the next line, if you click the Cancel button on the open file utility, the script is terminated. The **GETINPUT** command is used to allow you to enter the value to add to the type codes. The file is then opened. The variable "evcount" is created, and contains the number of events in the file obtained with the **GETEVENTCOUNT** command. The "for" line creates the "index" variable (set to 0), and then says to increment (**incr**) index as long as the index value is less than "evcount" (the number of events). The program will loop until the criterion is met.

For each pass through the next sequence, **EventType** will always be Stimulus for stimulus events (this prevents other types of events from being changed). The "stimcode" variables are created, based on the information obtained from the stimulus codes. The **SETEVENTINFO** command then replaces the **StimulusCode** with value obtained by the expression (**expr**): stimulus code plus the specified value to add.

The next section builds the output file name and path. The "Pickoff" commands are used to get the parts of the name/path, and the format command builds the output path/name. (%s is used for strings, and %d is used for integers). The \$OutPath file is the path\filename\_Incremented\_By\_number, where path, filename and number are all entered when the file is executed.

**Using the Same Code Repeatedly.** If you find yourself using the same code over and over again, such as in the following example,

```
set accepted 100
set rejected 50
set report [format "accepted: %d rejected: %d" $accept $reject]
INSTRUCT $report
set accepted 101
set rejected 51
set report [format "accepted: %d rejected: %d" $accept $reject]
INSTRUCT $report
```

consider using the "proc" command. Procedures are a way of implementing new commands as TCL scripts (see chapter 8 of the Ousterhout text). They can appear in a single script as in the following:

```
proc AccRejNotify {accept reject} {
    set report [format "accepted: %d rejected: %d" $accept
$reject]
    INSTRUCT $report
}

AccRejNotify 100 50 ;# the proc (or subroutine) can be called as
many
times as needed
AccRejNotify 101 51
```

You can build a "library" to use whenever you need it. NSlibrary.tcl contains only one procedure but it could contain many. It must be executed (ie "sourced") in any script so that the procedures become available. Using the "library", our script would simplify to

```
source {c:\scan4.3\Acquire4.3\NSLibrary.tcl} ;# make sure your
path is
correct
AccRejNotify 100 50
AccRejNotify 101 51
```

The NSlibrary.tcl looks like the following:

```
proc AccRejNotify {accept reject} {
    set report [format "accepted: %d rejected: %d" $accept
$reject]
    INSTRUCT $report
}
```

## Commands for ACQUIRE and EDIT

ADDDISPLAYFILTER  
ADDWAVEBOARD  
ARRANGEWINDOWS  
AUTOSCALE  
BUTTERFLYPLOT  
CHANLIST  
DELETE  
GETBASELINECOR  
GETCHANATTRIBUTE  
GETCHANLABEL  
GETCHANNELINDEX  
GETDISPLAYINVERTED  
GETDISPLAYPAGE  
GETINPUT  
GETINPUTFILE  
GETNUMCHANS  
GETNUMDISPLAYPAGES  
GETNUMPOINTS  
GETNUMSWEEPS  
GETOUTPUTFILE  
GETSWDISPLAYSPEED  
GETSWEEPMAX  
GETSWEEPMIN  
HERTZTOPOINT  
INSERT  
INSTRUCT  
ISOFFLINE / ISONLINE  
LATENCYTOPOINT  
MAXIMIZE  
MNIMIZE  
NEWNAME  
PAUSE  
PICKOFFDIRECTORY  
PICKOFFEXTENSION  
PICKOFFFILENAME  
PICKOFFNAMEONLY  
POINTTOHERTZ  
POINTTOLATENCY  
PRINTWAVEBOARD  
READPOS  
READSUB  
REM  
REMOVEDISPLAYFILTER

RENAME  
RESTORE  
REVIEW  
SAVEPOS  
SAVESUB  
SCALE  
SCALEX  
SETBASELINECOR  
SETCHANATTRIBUTE  
SETDISPLAYINVERTED  
SETDISPLAYPAGE  
SETSPECTDISPLAYSTYLE  
SETSWDISPLAYSPEED  
SUBJECT  
TOGGLEZOOM  
WRITELOG  
ZOOMIN  
ZOOMOUT

## Commands for ACQUIRE

CALIB  
CLEARAVG  
CONFIGUREBLINKNOISEREDUCTION  
CONFIGUREEKGNOISEREDUCTION  
DODCCORRECT  
DOIMPEDANCE  
ENABLEBLINKNOISEREDUCTION  
ENABLEEKGNOISEREDUCTION  
GETAST  
HIGHIMPEDANCE  
PAUSEDISPLAY  
RESTART  
STARTACQUISITION  
STARTRECORDING  
STOPACQUISITION  
STOPRECORDING  
VIEW

## Commands for EDIT

ADD  
ADDANNOTATION  
APPEND  
APPENDRECORDING



AREAREPORT  
ARTCOR  
ARTREJ  
ARTREJ\_EX  
ARTREJCNT  
ARTREJCNTFREQ  
ARTREJFREQ  
ARTREJFREQ\_EX  
AVERAGE  
AVERAGE\_EX  
AVGBANDS  
BASECOR  
BASECOR\_EX  
BLINKNOISEREDUCTION  
CLEARALLMARKERS  
CLEARART  
CLEARMARKERS  
CLOSEALL  
CLOSEFILE  
COHERENCE  
COHERENCE\_EX  
CONCATCNT  
CREATESORT  
CROSSCORR  
CROSSCORR\_EX  
CUTEPOCH  
DCCORRECT  
DECIMATE  
DELETEALLCOMP  
DELETEBADCHANNELS  
DELETESORT  
DELETESWEEPS  
DETREND  
DETREND\_EX  
EKGNOISEREDUCTION  
ENABLEOVERWRITEPROMPT  
EPOCH  
EPOCH\_EX  
ERBP  
ERCOH  
EXCLUDEFORBASECOR  
EXCLUDEFORDETREND  
EXCLUDEFORFILTER  
EXCLUDEFORPEAKDETECTION  
EXCLUDEFORSMOOTH  
EXCLUDEFORWINDOWDATA

EXPORTAVG  
EXPORTAVG\_EX  
EXPORTCNT  
EXPORTCNT\_EX  
EXPORTCOH\_EX  
EXPORTEDF  
EXPORTEEG  
EXPORTEEG\_EX  
EXTRACT  
FILTER  
FILTER\_EX  
FORWARDFFT  
FSP  
FSP\_EX  
GETCOMP  
GETCURRENTEPOCHINDEX  
GETEPOCHCOUNT  
GETEPOCHINFO  
GETEVENTCOUNT  
GETEVENTINFO  
GETMARKERCOUNT  
GETMARKERINFO  
GETPOINTDATA  
GFP  
GROUPAVG  
IMPORTAVG  
IMPORTCNT  
IMPORTEEG  
INSERTSTIMEVENT  
INSERTRESPONSEEVENT  
INTEGRATEWAVEFORM  
INTRACLASSCORR  
INTRACLASSCORR\_EX  
INVERSEFFT  
LDR  
LOAD3DD  
MERGE  
MERGEEVT  
OPENFILE  
PAIRED  
PEAKDETECTION  
PEAKDETECTION\_EX  
PEARSONS  
PEARSONS\_EX  
PRINT  
READREJ

RECTIFY  
RECTIFY\_EX  
REFER  
REJECTBLOCK  
RESETFORBASECOR  
RESETFORDETREND  
RESETFORFILTER  
RESETFORPEAKDETECTION  
RESETFORSMOOTH  
RESETFORWINDOWDATA  
RESPWIN  
SAVEAS  
SAVEPEAK  
SAVEREJ  
SAVEEVENT  
SAVEEVENT\_EX  
SELECTFILE  
SETCOHREF  
SETCURRENTEPOCHINDEX  
SETEPOCHINFO  
SETEVENTINFO  
SMOOTH  
SMOOTH\_EX  
SNR  
SORT  
SPATIALFILTER  
SPATIALSVD  
SPECTRUM  
SPLINEFIT  
SUBTRACT  
TSCORE  
UPDATECHANGES  
VOLTAGETHRESHOLD  
WINDOWDATA  
WINDOWDATA\_EX  
WRITENUMSWEEPS  
ZSCORE

## Discontinued DOS Commands







---

ACCURACY  
ACQUIRE  
AVGFILE  
AVGSAVE  
CARTESIAN  
CLEAR


CLOSEPEAK  
CNTAVG  
CNTFILE  
CNTSAVE  
DRAW  
EDIT  
EEGFILE  
EEGSAVE  
EPOCH\_OVERLAP  
ELECTSCORE  
EXPORTMAC  
FCARTOON  
FSP\_STOPMETHOD  
FSP\_STOPMODE  
GETAVG  
GETCLIP  
GETCNT  
GETEEG  
GETMAP  
GETPEAKTEMP  
GROUP  
HOLD  
INDIVIDUAL  
LARGEMAP  
LATECENCYRNG  
MAPGEN  
MAXSWEEPS  
MEANFREQ  
MINSWEEPS  
OPENCNT  
OPENEEG  
OPENPEAK  
PLAYMOV  
POLAR  
QUIT  
READEVENT  
RECORD  
RECORDMOV  
RENAMECLIP  
RESPCODE  
RESPONSERNRG  
RETAIN  
REVERT  
SAVEAVG  
SAVECOMP  
SAVECSA

SAVEDAT  
SAVESCREEEN  
SEARCH  
SEQUENCE  
SETART  
SETARTCORPARAM  
SETARTCORREVIEW  
SETARTCORASSIGN  
SETBAD  
SETBASE  
SETFILTRECTIFY  
SETREF  
SETSKIP  
SETSPECT  
STATS  
SYSTEM  
TAGONCLIP  
TAGOFFCLIP  
TASKFILE  
TCARTOON  
TRIALRNG  
TYPERNG  
VARIANCE  
VECTORSORT  
WAVEFORM  
WINDOW

# Quick Reference Guide for SCAN BATCH Commands

The following is an alphabetical list of all of the commands used in SCAN, with a brief description of the function and a list of the parameters used by the command. Following each command you will see the ACQUIRE , EDIT , or both icons , indicated in which mode(s) the commands can be used. You will also see icons (where relevant) indicating the types of files that may be used with the commands:  indicates CNT files,  indicates EEG files,  indicates AVG files.


Commands that have been added in SCAN 4.3 and 4.3.1 are designated as **NEW**.

**ADD** - Add.  The ADD command is used to perform a point by point mathematical addition of the selected AVG file to the one specified by the first parameter and create a new output file.


1	string	Filename to add
2	string	Output file name

**ADDANNOTATION** - Add Annotation.  This command is used to add an annotation to a CNT file at a designated point.


1	string	Annotation text
2	int	Point offset. (zero-based index)

**ADDDISPLAYFILTER** - Add Display Filter.  This command is used to add filters to the active data file (this affects the display only - not the stored data). Filtering is analog simulation (zero-phase not available). See also FILTER.

1	defined value	Filter type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
2	float	High pass cutoff
3	int	High pass dBs (6, 12, 24, or 48)
4	float	Low pass cutoff
5	int	Low pass dBs (6, 12, 24, or 48)
6	float	Notch start frequency
7	float	Notch stop frequency
8	int	Notch dBs (6, 12, 24, or 48)
9	Boolean	Rectify
10	list	Channels to be filtered (or "All")

**ADDWAVEBOARD** - Add Waveform(s) to Waveboard.  The ADDWAVEBOARD command will add a waveform from the designated electrode names to the Waveboard.

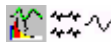
1	list	Electrode name(s) or "ALL"
---	------	----------------------------

**APPEND** - Append Channels.  The APPEND command inserts an additional channel(s) from the file specified by filename to the working file. This command can be used to build a new electrode array from a series of smaller separate recordings.


1	string	Input file
2	list	Channels to be included, or "All"
3	string	Output file

**APPENDRECORDING** - Append Recording.  APPENDRECORDING allows you to attach one or more CNT files together into a single CNT file. All files, with their paths, should be listed.


1	list	File(s) to append
2	string	Output file name

**AREAREPORT** - Area Report.  The AREAREPORT is used to create an ASCII file (DAT extension) that has the Sum, Mean or Area information for the interval specified. This replaces the AREA command from the DOS version.


1	string	Report file name
2	defined value	Computation type (SUM, MEAN, AREA)
3	Boolean	Rectify
4	Boolean	Use entire interval
5	double	Start latency (ignored if param 4 is TRUE)
6	double	End latency (ignored if param 4 is TRUE)

**ARRANGEWINDOWS** - Arrange Windows in the Display.  When multiple displays are open, this command is used to arrange the displays using the standard Windows conventions.

1	defined value	Method (CASCADE, TILEHORZ, TILEVERT)
---	---------------	--------------------------------------

**ARTCOR** - Ocular Artifact Reduction.  The ocular artifact reduction transform is used to subtract out blink or other regularly occurring artifacts (refer to the EDIT manual for details). It is used preferably with CNT files, but may be used with EEG files as well.



1	defined value	Trigger direction (POSITIVE, NEGATIVE)
2	double	Threshold percentage
3	int	Minimum sweeps
4	double	Sweep duration
5	string	Blink channel
6	defined value	Output switch (LDR, LDR+CNT)
7	string	LDR file name
8	string	Output CNT file (if param 6 is LDR+CNT)
9	Boolean	Review maxima (CNT files only)
10	Boolean	Review blinks (CNT files only)

**ARTREJ** - Artifact Rejection.  The Artifact Rejection option will automatically reject sweeps in which the voltage in a designated channel(s) exceeds defined criteria. The GETCHANATTRIBUTE / SETCHANATTRIBUTE commands can be used to set which channels will be used for artifact rejection.



1	defined value	Operation type (CRITERIA or REJCRITERIA*, REJECTALL, ACCEPTALL, ACCRCRITERIA)
2	Boolean	Use entire interval
3	double	Start reject interval (ignored if parameter 2 is TRUE)
4	double	Stop reject interval (ignored if parameter 2 is TRUE)
5	Boolean	Re-compute
6	double	Minimum amplitude
7	double	Maximum amplitude

8	Boolean	Exclude bad channels
9	Boolean	Exclude skipped channels

\* In SCAN 4.3, the "CRITERIA" option was available. In SCAN 4.3, the "REJCRITERIA" term was added to balance the new "ACCCRITERIA" option. In SCAN 4.3 the program will recognize CRITERIA as it did in SCAN 4.3, as well as REJCRITERIA - the program interprets the two equivalently.



**ARTREJ\_EX** - **NEW** Artifact Rejection, extended.   The ARTREJ\_EX command is the same as the ARTREJ command, except that it allows you to specify which channels should be the artifact rejection channels and the options to exclude bad or skipped channels are removed (since they are really not needed when you select the channels to include).

1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	Boolean	Use entire interval
3	double	Start reject interval (ignored if param 2 is YES)
4	double	Stop reject interval (ignored if param 2 is YES)
5	Boolean	Recompute
6	double	Minimum amplitude
7	double	Maximum amplitude
8	Boolean	Exclude Bad channels
9	Boolean	Exclude Skipped channels
10	list	List of artifact rejection channels

**ARTREJCNT** - **NEW** Artifact Rejection for CNT files (time domain correction).   The routine scans all, or specified channels for voltages exceeding the criteria set in the Min and Max parameters. When a voltage exceeds either criterion, a block of data is rejected according to the limits set in the Block Interval parameters. The interval of the rejected block is defined by the pre-Artifact and Post-Artifact time spans. The Refractory period is a span of time following the artifact during which additional artifacts will not be detected. (Please see the EDIT manual for more details).

1	double	Pre-artifact time span (ms)
2	double	Refractory period (ms)
3	double	Post-artifact time span (ms)
4	Boolean	Recompute
5	double	Min amplitude (uV)
6	double	Max amplitude (uV)
7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skip channels
9	list	List of rejection electrodes



**ARTREJCNTFREQ** - **NEW** Artifact Rejection for CNT files (frequency domain correction).

  This routine performs a series of FFTs, and then rejects epochs when the power in a specified range exceeds a specified voltage threshold. The span of the epochs is determined by the Epoch Points parameter, and that will interact with the AD rate used when the file was recorded.

1	integer	Epoch Points (power of 2)
2	double	Slide interval duration (ms)
3	double	Start frequency
4	double	Stop frequency
5	double	Threshold (uv)
6	Boolean	Recompute





7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skipped channels
9	list	List of electrodes to include




**ARTREJFREQ** - **NEW** Artifact Rejection based on frequency band amplitude.   The ARTREJFREQ command uses the Frequency Domain criteria to accept/reject sweeps. An FFT is computed (whether the numbers of points is a power of 2 or not), and sweeps are then accepted or rejected on the basis of the amplitude criteria you select within a specified band width. The GETCHANATTRIBUTE / SETCHANATTRIBUTE commands can be used to set which channels will be used for artifact rejection.



1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	double	Start reject frequency
3	double	Stop reject frequency
4	Boolean	Recompute
5	double	Maximum amplitude
6	Boolean	Exclude Bad channels
7	Boolean	Exclude Skipped channels

**ARTREJFREQ\_EX** - **NEW** Artifact Rejection based on frequency band amplitude, extended.

  The ARTREJFREQ\_EX command is similar to the ARTREJFREQ command except that it allows you to specify which channels should be the artifact rejection channels, and the options to exclude bad or skipped channels are removed (since they are really not needed when you select the channels to include).



1	defined value	Operation type (REJCRITERIA, REJECTALL, ACCEPTALL, ACCCRITERIA)
2	double	Start reject frequency
3	double	Stop reject frequency
4	Boolean	Recompute
5	double	Maximum amplitude
6	list	List of artifact rejection channels

**AUTOSCALE** - Autoscale.    The AUTOSCALE command will automatically search for the minimum and maximum values in the electrode array. The display min and max values will be set to the found minimum and maximum values, respectively. Electrodes set to be skipped will not be included in the search.



**AVERAGE** - Average Sweeps.   The AVERAGE command is used to average a sequence of single sweeps (EEG file) into an average file (AVG file) of all the selected sweeps. See also CREATSORT. In SCAN 4.3, SNR is computed (for time domain files only) using the prestimulus interval for the Noise estimate (or the 20th percentile if there is no prestim interval), and the poststimulus interval for the Signal estimate.

1	defined value	Domain (TIME, FREQUENCY)
2	Boolean	Compute standard deviation
3	Boolean	Compute SNR
4	string	SNR filename
5	defined value	Spectral scaling method (AMPLITUDE, POWER)
6	int	Spectral window length (Taper%)
7	defined value	Spectral window type (COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)



8	string	Sort name
9	string	Output file name

**AVERAGE\_EX** - **NEW** Average Sweeps, Extended.   This command is the same as the AVERAGE command except for the way in which the SNR is computed. You have control over the options that used (for the intervals; see the EDIT manual for more details).



1	defined value	Domain (TIME, FREQUENCY)
2	Boolean	Compute standard deviation
3	defined value	Spectral scaling method (AMPLITUDE, POWER)
4	int	Spectral window length (Taper %)
5	defined value	Spectral window type (COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
6	defined value	Noise interval Type (PRESTIMINTERVAL, PERCENTILE, USERDEFINED)
7	float	SNR Noise Start
8	float	SNR Noise End
9	defined value	Signal interval Type (POSTSTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
10	float	SNR Signal Start
11	float	SNR Signal End
12	string	Sort name
13	string	Output file name

**AVGBANDS** - Modify Frequency Bands in AVG Files.   (frequency domain AVG and COH files only). This option formerly existed as the AVGBAND.exe utility program for the DOS version of EDIT. It is used to redefine the frequency bands that are exported to an ASCII file. The output is a text file with a .bnd extension. Repeat the label/start/stop sequence for as many bands as you want to create.

1	string	Output band file name
x	string	Band label
x	double	Band start frequency
x	double	Band stop frequency



**BASECOR** - Baseline Correction.   The baseline correct command allows you to modify the current DC offset of the waveform. See also EXCLUDEFORBASECOR and RESETFORBASECOR.

1	defined value	Interval type (FIRSTPOINT, PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start latency (if Interval type is USERDEFINED)
3	double	Interval stop latency (if Interval type is USERDEFINED)
4	Boolean	Exclude skipped channels
5	Boolean	Exclude bad channels
6	string	Output file name (EEG files only)



**BASECOR\_EX** - **NEW** Baseline Correction, extended.   This command differs from the BASECOR command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Interval type (FIRSTPOINT, PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start latency (only if 2 is USERDEFINED)


3	double	Interval stop latency (only if 2 is USERDEFINED)
4	list	List of electrodes to correct, or "All"
5	string	Output file (not needed for AVG files)



**\*BLINKNOISEREDUCTION** -  **NEW** Blink Noise Reduction Routine.  The BLINKNOISEREDUCTION command is intended for the removal of VEOG blink artifact from CNT files (see the EDIT manual for details). See also the **ARTCOR** command for an alternate (but similar) way to remove blink artifact.

1	string	Output file
2	double	Epoch start time (ms)
3	double	Epoch stop time (ms)
4	integer	Number of averages
5	Boolean	Use external trigger
6	double	Voltage threshold (uVs; ignored if #5 is true)
7	defined value	Trigger direction (Positive, Negative; ignored if #5 is true)
8	string	Artifact channel label (REQUIRED even if #5 is true)
9	int	Trigger code (ignored if #5 is false)
10	Boolean	Enable artifact rejection
11	double	Artifact min (uV)
12	double	Artifact max (uV)
13	Boolean	Insert events
14	int	Event code
15	Boolean	Enable bipolar reference
16	string	Bipolar reference channel label
17	Boolean	Enable high pass filter
18	double	High pass filter value

**BUTTERFLYPLOT** -  **NEW** Show as Butterfly Plot.  This command is used to display the time domain AVG and EEG files in the Butterfly Plot style.



1	Boolean	Enable butterfly plot
---	---------	-----------------------



**CALIB** - Calibration.  The CALIB command will perform an automatic calibration procedure and display the calibration values. You have the option to accept or reject them.


**CHANLIST** - Channel List.   This command will display the standard montage screen, from which you may view or modify the selected electrodes.


**Return.** A list containing all of the selected channels.

1	list	Starting list of selected electrodes
2	string	Title


**CLEARALLMARKERS** - Clear All Markers.   This command will remove markers that have been created with the Add Marker option (right mouse menu for Multiple Windows displays) or the PEAKDETECTION command. All markers will be removed from all channels across all epochs (or the single epoch with AVG files). See also **CLEARMARKERS**.


**CLEARART** - Clear Artifacts.   The CLEARART command will reset all single-sweep accept/reject status of all epochs in an EEG file to the accept state. This command is provided for compatibility with earlier versions. The same result may be more easily implemented with the Clear All parameter in the ARTREJ command.

**CLEARAVG** - Clear Average.  This command is used to clear the accumulating average online, and to restart the average with the next accepted sweep. The command is not applicable when you have multiple online sorted averages - just the single average you see when recording in Average Acquisition mode.


**CLEARMARKERS** - Clear Markers.  This command is used to remove markers that have been created with the Add Marker option (right mouse menu for Multiple Windows displays) or the PEAKDETECTION command. With EEG files, markers will be removed from the displayed sweep only (use CLEARALLMARKERS to remove all of them).

1	list	Selected channels, or All
---	------	---------------------------


**CLOSEALL** - Close All Files.  The CLOSEALL command will close all files that you have opened. You will not be prompted to save modified data files. If the transform you are using requires a separate output file, the SAVEAS command may be used to save the file with a different name before closing. See Also OPENFILE and CLOSEFILE.

**CLOSEFILE** - Close File.  The CLOSEFILE command is used to close a specified file. Specify the filename and the extension without a path (e.g., Viscpt.cnt). If you use CLOSEFILE, you will not be prompted to save modified data files. If the transform you are using requires a separate output file, the SAVEAS command may be used to save the file with a different name before closing. See also OPENFILE and CLOSEALL.


1	string	File name to close (no path)
---	--------	------------------------------

**COHERENCE** - Coherence.  The Coherence command will perform analyses on EEG files where the number of points must be a power of 2 (256, 512, etc.). Use the SPLINEFIT command if needed to get the correct number of points. (See the Coherence section of the EDIT manual for more details about the options with Coherence).

1	double	Computation of maximum frequency
2	defined value	Pair selection state (ALLPAIRS, MNTFILE)
3	string	MNT file name (ignored if parameter 2 is ALLPAIRS)
4	Boolean	Spectral covariance
5	Boolean	Retain mean
6	string	Output file name


**COHERENCE\_EX** - **NEW** Coherence, extended.  The COHERENCE\_EX command is the same as the prior COHERENCE command, with the addition of a sorting option (new parameter 6).

1	double	Computation of maximum frequency
2	defined value	Pair selection state (ALLPAIRS, MNTFILE)
3	string	MNT file name (ignored if param 2 is all pairs)
4	Boolean	Spectral covariance
5	Boolean	Retain mean
6	string	Sort name
7	string	Output file name (or "")

**CONCATCNT** - Concatenate CNT files.  This is another way to combine CNT files aside from APPENDRECORDING. The difference is that you do not need to use OPENFILE first - just specify the files to be appended, and the output file name.


1	list	File(s) to append ("" will fail)
2	string	Output file name (or "")

**\*CONFIGUREBLINKNOISEREDUCTION** - **NEW** Configure Blink Noise Reduction


Routine.  The CONFIGUREBLINKNOISEREDUCTION option is used to remove VEOG blink artifact during the acquisition of CNT files. See also ENABLEBLINKNOISEREDUCTION. The same reduction may be applied offline; see BLINKNOISEREDUCTION.

1	double	Epoch start time (ms)
2	double	Epoch stop time (ms)
3	integer	Number of averages
4	Boolean	Use external trigger
5	double	Voltage threshold (uV; ignored if #4 is true)
6	defined value	Trigger direction (Positive, Negative; ignored if #4 is true)
7	string	Artifact channel label (ignored if #4 is true)
8	int	External trigger code (ignored if #4 is false)
9	Boolean	Enable artifact rejection
10	double	Artifact min (uV)
11	double	Artifact max (uV)
12	Boolean	Enable bipolar reference
13	string	Bipolar reference channel label
14	Boolean	Enable high pass filter
15	double	High pass filter value


**\*CONFIGUREEKGNOISEREDUCTION** - **NEW** Configure EKG Noise Reduction

Routine.  The CONFIGUREEKGNOISEREDUCTION option is used to remove ballistocardiogram or other EKG artifact during the acquisition of CNT files. See also ENABLEEKGNOISEREDUCTION. The same reduction may be applied offline; see EKGNOISEREDUCTION.


1	double	Epoch start time (ms)
2	double	Epoch stop time (ms)
3	integer	Number of averages
4	Boolean	Use external trigger
5	double	Voltage threshold (uV; ignored if #4 is true)
6	defined value	Trigger direction (Positive, Negative; ignored if #4 is true)
7	string	Artifact channel label (ignored if #4 is true)
8	int	External trigger code (ignored if #4 is false)
9	Boolean	Enable artifact rejection
10	double	Artifact min (uV)
11	double	Artifact max (uV)
12	Boolean	Correct trigger channel
13	Boolean	Enable bipolar reference
14	string	Bipolar reference channel label
15	Boolean	Enable high pass filter
16	double	High pass filter value

**CREATESORT** - Creates a Sort.  See the CREATESORT command for complete details, as well as DELETESORT.

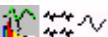
1	string	Sort name
---	--------	-----------

**CROSSCORR** - **NEW** Cross Correlation.  The Cross Correlation coefficient statistic is created by computing the correlation between electrodes across a lag series. It can be used to examine the relationships between or among electrodes by shifting the waveforms in time, and then recomputing the correlations (see the EDIT manual for more details). Retrieve one file, and then use the CROSSCORR command.


1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval stop
4	string	Output data file name (or "")

**CROSSCORR\_EX** - **NEW** Cross Correlation, extended.  The CROSSCORR\_EX command is the same as the CROSSCORR command, except for the new 5th parameter that lets you specify the channels to be included.



1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval stop
4	string	Output data file name (or "")
5	list	Channels to compute

**CUTEPOCH** - Cut Epoch.  The Cut Epoch option will create a new AVG or EEG file with Start and End time times specified. **Important: the start and end time must be within the original epoch interval.**



1	double	Start latency
2	double	End latency
3	string	Output file name

**DCCORRECT** - DC Correction.  The DC Correction option will attempt to correct for DC offsets and DC drifting in continuous files recorded on a SynAmps in DC mode with a DC High Pass filter (see the EDIT manual for complete details).



1	double	Pre-correction start time
2	double	Pre-correction stop time
3	double	Post correction start time
4	double	Post-correction stop time
5	Boolean	Perform DC drift correction
6	Boolean	Save drift waveforms
7	int	Polynomial order
8	Boolean	Exclude stimulus events
9	Boolean	Exclude response events
10	Boolean	Exclude keyboard events
11	double	Exclude interval start time
12	double	Exclude interval stop time
13	string	Drift file name (ignored if save drift is FALSE)
14	Boolean	Use interval mode

**DECIMATE** - **NEW**.   Use this option to decimate CNT files to a lower AD rate. Retrieve the CNT file, select the Decimate transform, and select a New Sample rate from the list of pull-down options. Enter an output file name, and click OK. A new CNT file will be created with the lower AD rate (and smaller file size). Prior to decimation, an IIR low pass filter is applied (passed four times) to correct for aliasing.

1	int	New AD Rate
2	string	Output file name

**DELETE** - Delete File(s).   This command can be used to delete a file(s) without exiting the program. The filename should be specified with an extension and a full path.



1	list	File name(s)
---	------	--------------

**DELETEBADCHANNELS** - Delete Bad Channels.   The DELETEBADCHANNELS command is used to create a new continuous file with the "Bad" channels excluded. Use the SETCHANATTRIBUTE command to designate a channel as being Bad, if desired.



1	string	Output file name
---	--------	------------------

**DELETESORT** - Deletes a Sort.  See also CREATESORT.



1	string	Sort name
---	--------	-----------

**DELETESWEEPS** - Delete Rejected Sweeps.   This command will create a new epoched (EEG) file with the rejected sweeps removed. Use the ARTREJ command to reject the sweeps automatically, then create the new file, if desired.

1	string	Output file name
---	--------	------------------



**DETREND** - Linear Detrend.   This option is used to remove linear drifting components (such as, HEOG artifact).

1	defined value	Interval type (PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start (ignored if parameter 1 is not USERDEFINED)
3	double	Interval end (ignored if parameter 1 is not USERDEFINED)
4	string	Output file name (for EEG Files only)



**DETREND\_EX** - **NEW** Detrending, extended.   This command differs from the DETREND command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	interval type (PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval Start (ignored if parameter 1 is not USERDEFINED)
3	double	Interval End (ignored if parameter 1 is not USERDEFINED)
4	list	Channels to detrend
5	string	output file (only for EEG files)



**DODCCORRECT** - Do a DC Correction.   This command is used to perform a planned DC correction (SynAmps only, with a DC high pass filter setting). You might wish to use this command at the beginning of a recording, or prior to Impedance testing, to remove any DC offset or drifting.

**DOIMPEDANCE** - Do Impedance Testing.  This command will initiate impedance testing.

**\*EKGNOISEREDUCTION** - **NEW** EKG Noise Reduction Routine.   The EKGNOISEREDUCTION command is intended for the removal of ballistocardiogram and other EKG artifact from CNT files (see the EDIT manual for details).

1	string	Output file
2	double	Epoch start time (ms)
3	double	Epoch stop time (ms)
4	integer	Number of averages
5	Boolean	Use external trigger
6	double	Voltage threshold (uV; ignored if #5 is true)
7	defined value	Trigger direction (Positive, Negative; ignored if #5 is true)
8	string	Artifact channel label (ignored if #5 is true)
9	int	External trigger code (ignored if #5 is false)
10	Boolean	Enable artifact rejection
11	double	Artifact min (uV)
12	double	Artifact max (uV)
13	Boolean	Correct trigger channel
14	Boolean	Insert events
15	int	Event code
16	Boolean	Use external average
17	string	External AVG file (ignored if 16 is false)
18	Boolean	Enable bipolar reference
19	string	Bipolar reference channel label
20	Boolean	Enable high pass filter
21	double	High pass filter value

**\*ENABLEBLINKNOISEREDUCTION** - **NEW** Enable Blink Noise Reduction Routine.




  The ENABLEBLINKNOISEREDUCTION command is used to initiate the reduction routine. The routine is configured using CONFIGUREBLINKNOISEREDUCTION.

1	Boolean	Enable the routine
---	---------	--------------------

**\*ENABLEEKGNOISEREDUCTION** - **NEW** Enable EKG Noise Reduction Routine.


  The ENABLEEKGNOISEREDUCTION command is used to initiate the reduction routine. The routine is configured using CONFIGUREEKGNOISEREDUCTION.

1	Boolean	Enable the routine
---	---------	--------------------


**ENABLEOVERWRITEPROMPT** - Overwrite Existing File Prompt.    This command is used to display or not display the message asking whether you wish to overwrite an existing data file. Disabling the command will allow the BATCH program to run without user intervention (be sure that you do not unintentionally overwrite files you want to keep).

1	Boolean	Enable overwrite prompt
---	---------	-------------------------




**EPOCH** - Epoch.  The EPOCH command will transform a continuous (\*.CNT) file into an epoched or single-sweep file (\*.EEG). See also **CREATESORT**.

1	defined value	Trigger mode (PORT_INTERNAL, NOTRIGGER, EVENTFILE)
2	string	Event file name (ignored unless #1 is EVENTFILE)
3	double	Start latency
4	double	Stop latency
5	Boolean	Response locked
6	Boolean	Reject epochs that overlap rejected blocks
7	Boolean	Include stimulus events
8	Boolean	Include keyboard events
9	Boolean	Include response pad events
10	string	Sort name
11	string	Output file


**EPOCH\_EX** - **NEW** Epoch, Extended.  The EPOCH\_EX command differs from the original EPOCH command in that it will accept an event file that is seconds (rather than byte or point offsets).

1	defined value	Trigger mode (PORT_INTERNAL, NOTRIGGER, EVENTFILE)
2	string	Event file name (ignored unless #1 is EVENTFILE)
3	Boolean	Event file is in seconds (ignored unless #1 is EVENTFILE)
4	double	Start latency
5	double	Stop latency
6	Boolean	Response locked
7	Boolean	Reject epochs that overlap rejected blocks
8	Boolean	Include stimulus events
9	Boolean	Include keyboard events
10	Boolean	Include response pad events
11	string	Sort name
12	string	Output file


**ERBP** - Event Related Band Power.  This transform computes power (or amplitude) of induced (and/or evoked) event-related EEG activity in a centered frequency band as a function of time. The options that are available for some parameters are dependent on the selections in some other parameters. For example, if you select POWER for #1, the AMPLITUDE option is ignored in #15. If #10 is not enabled, #11 and #12 will be ignored. There are several other examples. We strongly encourage you to use the ERBP window in Point & Click mode to set up the transform, and then transfer those settings to your BATCH file. See also **CREATESORT**.

1	defined value	Method (POWER, AMPLITUDE)
2	Boolean	Enable bandpass
3	defined value	Filter Mode (ZEROPHASESHIFT, ANALOGSIMULATION)
4	double	Center frequency (Hz)
5	double	Half bandwidth (Hz)
6	int	Bandpass attenuation (dB)
7	Boolean	Enable Envelope
8	Boolean	Warm up from right
9	defined value	Phase locking (INDUCED, EVOKED, BOTH)
10	Boolean	Enable reference interval
11	double	Reference start (ms's)
12	double	Reference stop (ms's)


13	double	Trim left (ms's)
14	double	Trim right (ms's)
15	defined value	Scaling (POWER, AMPLITUDE, PERCENT, Z)
16	Boolean	Enable Averaging window
17	double	Averaging window (ms's; ignored if #16 is FALSE)
18	defined value	Window treatment (COLLAPSE, SMOOTH)
19	string	Sort name
20	string	Output file

**ERCOH** - Event Related Coherence.  ERCoH is computed from epoched EEG data using the coherence formulas already given in the EDIT manual in the Coherence section. However, in this case, the frequency of interest is pre-selected, and the results are a function of time with respect to the event at time zero (refer to the EDIT manual for more details). The options that are available for some parameters below are dependent on the selections in some other parameters. For example, if you select COHERENCE in #1, the EVOKED option is ignored in #6. If you select COVARIANCE in #1, EVOKED and BOTH are ignored in #6. If you select CROSSPRODUCT in #1, INDUCED is ignored in #6. It would be a good idea to use the ERCOH window in Point & Click mode to set up the transform, and then transfer those settings to your BATCH file. See also CREATSORT.


1	defined value	Type (COHERENCE, COVARIANCE, CROSSPRODUCT)
2	double	Center frequency (Hz)
3	double	Half bandwidth (Hz)
4	int	Bandpass attenuation (dB)
5	Boolean	Warm up from right
6	defined value	Phase locking (INDUCED, EVOKED, BOTH)
7	double	Trim left
8	double	Trim right
9	defined value	Pair selection (ALLPAIRS, MNTFILE)
10	string	MNT file
11	string	Sort name
12	string	Output file (.coh extension)

**EXCLUDEFORBASECOR** - Exclude Channel(s) for Baseline Correction.  Use this command to exclude channels from Baseline Correction. See also BASECOR, BASECOR\_EX, RESETFORBASECOR.


1	list	Electrode name(s)
---	------	-------------------

**EXCLUDEFORDETREND** - Exclude Channel(s) for Linear Detrend.  Use this command to exclude channels from Detrending. See also DETREND and RESETFORDETREND.

1	list	Electrode name(s)
---	------	-------------------

**EXCLUDEFORFILTER** - Exclude Channel(s) for Filtering.  Use this command to exclude channels from Filtering. See also FILTER and RESETFORFILTER.


1	list	Electrode name(s)
---	------	-------------------

**EXCLUDEFORPEAKDETECTION** - Exclude Channel(s) for Peak Detection.  Use this command to exclude channels from Peak Detection. See also PEAKDETECTION and RESETFORPEAK-DETECTION.

1	list	Electrode name(s)
---	------	-------------------

**EXCLUDEFORSMOOTH** - Exclude Channel(s) for Smoothing.  Use this command to exclude channels from Smoothing. See also SMOOTH and RESETFORSMOOTH.


1	list	Electrode name(s)
---	------	-------------------

**EXCLUDEFORWINDOWDATA** - Exclude Channel(s) for Windowing.  Use this command to exclude channels from Windowing. See also WINDOWDATA and RESETFORWINDOWDATA.

1	list	Electrode name(s)
---	------	-------------------

**EXPORTAVG** - Export AVG File to ASCII.  This command allows you to export averaged time or frequency domain data to an ASCII file. See also IMPORTAVG.

1	string	Output file name
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include electrode labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include standard deviation
8	Boolean	Include Bad channels
9	Boolean	Include Skipped channels



**EXPORTAVG\_EX** - **NEW** Export AVG files, Extended.  This command allows you to export AVG files with the additional options to include Data Labels and Maximum Resolution (to additional decimal places). The Data Label is the line: [Average Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also.

1	string	Output file name (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include electrode labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include standard deviation
8	Boolean	Include Bad channels
9	Boolean	Include Skipped channels
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	list	List of channels to be exported (or ALL)



**EXPORTCNT** - Export CNT File to ASCII.  This command allows you to export a CNT file to an ASCII file. See also IMPORTCNT.

1	string	Output file
2	Boolean	Include header

3	Boolean	Include labels
4	Boolean	Include X units
5	Boolean	Include Y units
6	Boolean	Include bad channels
7	Boolean	Include skipped channels

**EXPORTCNT\_EX** - **NEW** Export CNT File to ASCII, Extended.   This command allows you to export a CNT file to an ASCII file, with the additional options to include the Data Labels and to use Maximum Resolution. The Data Label is the line: [Continuous Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also.

1	string	Output file
2	Boolean	Include header
3	Boolean	Include labels
4	Boolean	Include X units
5	Boolean	Include Y units
6	Boolean	Include bad channels
7	Boolean	Include skipped channels
8	Boolean	Include data labels
9	Boolean	Use maximum resolution
10	Boolean	Append if file exists
11	list	List of channels to be exported (or ALL)

**EXPORTCOH\_EX** - **NEW** Export COH files, Extended.   This command allows you to export COH files to ASCII. If you want to limit the pairs that are exported, you should use an MNT file when you execute the COHERENCE transform.



1	string	Output file (or "")
2	defined value	Method (POINTS, PAIRS)
3	Boolean	Include header
4	Boolean	Use maximum resolution
5	Boolean	Append if file exists

**EXPORTEDF** - **NEW** Export CNT File in European Data Format (EDF).   This command allows you to export a CNT file in European Data Format.



1	string	Output file (or "")
2	Boolean	Include bad channels
3	Boolean	Include skipped channels

**EXPORTEEG** - Export EEG File to ASCII.   This command allows you to export an EEG file to an ASCII file. See also IMPORTEEG.



1	string	Output file
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include bad channels
8	Boolean	Include skipped channels
9	Boolean	Include epoch headers

**EXPORTEEG\_EX** - **NEW** Export EEG File to ASCII, Extended.   This command allows you to export a EEG file to an ASCII file, with the additional options to include the Data Labels and to use Maximum Resolution (maximum number of decimal places). The Data Label is the line: [Epoch Data]. If you exclude the header, electrode labels, and X and Y units, this is the only line that is left in the ASCII file (aside from the data). If you want numerical data only in the output file, exclude the Data Label also.



1	string	Output file (or "")
2	defined value	Method (POINTS, ELECTRODES, BESA)
3	Boolean	Include header
4	Boolean	Include labels
5	Boolean	Include X units
6	Boolean	Include Y units
7	Boolean	Include bad channels
8	Boolean	Include skipped channels
9	Boolean	Include epoch headers
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	list	List of channel to be exported (or ALL)

**EXTRACT** - Extract Channels.   The **EXTRACT** command creates a file containing data from selected channels from the current AVG or EEG file. The electrode arguments must match electrode labels in the current file. See also **APPEND**.

1	list	Channels to be included
2	string	Output file name


**FILTER** - Filter.   Use this command to perform a variety of filtering transforms on your data file. See also **EXCLUDEFORFILTER** and **RESETFORFILTER**.

1	defined value	Filter type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
2	defined value	Filter mode (ZEROPHASESHIFT, ANALOGSIMULATION)
3	float	High pass cutoff
4	int	High pass dBs (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)
5	float	Low pass cutoff
6	int	Low pass dBs (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)
7	float	Notch start frequency
8	float	Notch stop frequency
9	int	Notch dBs (zero-phase filtering: 12, 24, 48, or 96; analog simulation: 6, 12, 24, or 48)
10	Boolean	Rectify
11(optional)	string	Output file name, or "" (omitted for AVG files)


**FILTER\_EX** - **NEW** Filtering, extended.   This command differs from the **FILTER** command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	defined value	Filter Type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
2	defined value	Filter Mode (ZEROPHASESHIFT, ANALOGSIMULATION)


3	float	High Pass cutoff
4	int	High Pass attenuation (dB)
5	float	Low Pass cutoff
6	int	Low Pass attenuation (dB)
7	float	Notch start frequency
8	float	Notch stop frequency
9	int	Notch attenuation (dB)
10	Boolean	Rectify
11	list	List of electrodes to filter or "All"
12(opt)	string	Output file name, or "" (omitted for AVG files)

**FORWARDFFT** - Forward FFT.  The **FORWARDFFT** command performs an FFT on epoched files. The number of points in each channel and sweep must be a power of 2 (256, 512, etc.). Use the **INVERSEFFT** command to restore the data.


1	string	Output file name
---	--------	------------------

**FSP** - Fsp Average.  The **Fsp Average** command performs a signal to noise (SNR) computation on epoched files as a means for helping to determine when a signal has emerged with statistical significance from the noise background (refer to the **EDIT** manual for complete details). See also **CREATESORT**.




1	defined value	Terminate method (NONE, FVALUE, NOISELEVEL, BOTH)
2	double	F value
3	double	Noise level
4	double	Single point position
5	int	Sweeps per block
6	double	Window start
7	double	Window stop
8	string	Sort name
9	string	Output file name

**FSP\_EX** - **NEW** Fsp Averaging, extended.  This command differs from the **FSP** command in that it allows you to list the electrodes that you want to include for the termination channels (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "**\_EX**" command automatically using the electrodes you specify in the "Select" dialog box (or "All").






1	defined value	Terminate Method (NONE, FVALUE, NOISELEVEL, BOTH)
2	double	F Value,
3	double	Noise Level,
4	double	Single Point Position
5	int	Sweeps Per Block
6	double	Window Start
7	double	Window Stop
8	string	Sort Name
9	list	List of terminate electrodes or "All"
10	string	Output file

**GETAST** - Get Setup File.  Use this command to load an AST setup file in **ACQUIRE**. If the name is omitted, you will be prompted to enter one.

1(optional)	string	AST file name
-------------	--------	---------------






**GETBASELINECOR** - Get Baseline Display Correction.    Used to query the baseline correction (display only) status of the current working file. See also SETBASELINECOR.

**Return.** The GETBASELINECOR command will return either a 0 (no correction applied), or a 1 (baseline correction applied).

**GETCHANATTRIBUTE** - Get Channel Attribute.      This command is used to return the channel attributes for a specified channel. See also SETCHANATTRIBUTE.






1	string	Channel label
2	defined value	Attribute Type (-Artifact, -Fsp, -Hide, -Skip, -Bad, -AutoAdd, AutoAddLast)
-		

**Return.** The command will return a 1 if the attribute is set, or a 0 if it is not.

**GETCHANLABEL** - Get Channel Label.      This command is used to retrieve a selected channel label. See also GETCHANINDEX.




1	int	channel index (zero-based index)
---	-----	----------------------------------

**Return.** The command will return the channel label.




**GETCHANNELINDEX** Get Channel Index.      This command is used to retrieve a 0 based channel index from a channel label. See also SETCHANLABEL.

1	string	electrode label
---	--------	-----------------






**Return.** 0 based index of the channel

**GETCOMP/ DELETEALLCOMP** - Get/Delete Comparison File(s).    Use the GETCOMP command to open a comparison AVG file and superimpose it on a previously retrieved working file. (The files must match in terms of electrode labels, number of channels, number of points, and latency/frequency start/stop times). Use the DELETEALLCOMP command to delete the comparison file(s) (no parameters).

1	string	Comparison file name
---	--------	----------------------



**GETCURRENTEPOCHINDEX** - Get Current Epoch Index.    This command will return the sweep number of the currently displayed epoch. GETCURRENTEPOCHINDEX uses no parameters. See also SETCURRENTEPOCHINDEX.

**Return.** The sweep number of the currently displayed epoch is returned.



**GETDISPLAYINVERTED** - Get Display Inversion.      This command is used to determine the current status of the Invert Polarity option of the current working file. See also SETDISPLAYINVERTED.

**GETDISPLAYPAGE** - Get Display Page.      This command is used to return the number of the current display page. See also SETDISPLAYPAGE.

**Return.** The current display page is returned.

**GETEPOCHCOUNT** - Get Epoch Count.   This command will return the number of epochs in an EEG file.

**Return.** The number of sweeps in the file is returned.

**GETEPOCHINFO** - Get Epoch Information.   These commands will return information from EEG files regarding specified epochs.



1	int	Epoch number (zero-based index)
2	defined value	Parameter (-TrialType, -Accept, -Correct, -ReactionTime, -Response)

**Return.** The specified information is returned. The returned values depends on which information is being requested

-TrialType	Returns the stimulus type code number
-Accept	0 = Reject, 1 = Accept
-Correct	0 = Incorrect, 1 = Correct
-ReactionTime	Returns the latency of the response
-Response	Returns the response type code number

**GETEVENTCOUNT** - Get number of events.   This command will return the total number of events in the CNT file (only) event table.

**Return.** The total number of events is returned.


**GETEVENTINFO** - Get Event Information.   This command will return information from the CNT file (only) event table

1	int	Event number (zero-based index)
2	defined value	Parameter (-EventType, -Offset, -ResponseLatency, -Accuracy, -KeyboardCode, -KeypadCode, -StimulusCode)

Depending upon which Parameter you selected for #2, the program will return differing information. The following translates the meaning of the returned codes.

-EventType	Stimulus, Keypad, Rejected sweep, Accepted sweep, Keyboard, DC Correction, and Segment (Stop/Start event)
-Offset	Number of points since the beginning of the file.
-ResponseLatency	The response latency in ms.
-Accuracy	NORESPONSE, INCORRECT, CORRECT
-KeyboardCode	Function key number (2-11)
-KeypadCode	Number of response pad button pressed (1-4)
-StimulusCode	The stimulus type code number (1-255)

**Return.** The returned value may be a number or a text string.

**GETINPUT** - Get Input Text.  This command can be used to create a simple dialog box for entering text that is then used in the BATCH program.

1	string	Title
2(optional)	string	Prompt




3(optional)	string	Edit default
-------------	--------	--------------

**Return.** The text entered by the user.

**GETINPUTFILE** - Get Input File.  This command is used to display the standard Open File utility display, with optional customization. See also GETOUTPUTFILE.


1	string	String title
2(optional)	string	Extension
3(optional)	string	Initial Path

**Return.** The full path of the file selected by the user.

**GETMARKERCOUNT** - **NEW** Get Marker Index Count.  This command returns the number of markers at a selected channel, and at a selected sweep. With AVG files, the 2nd parameter is ignored.


1	string	Channel label
2	int	Epoch index (zero-based; ignored for AVG files)

**Return.** The number of Markers is returned.


**GETMARKERINFO** - **NEW** Get Marker Information.  The GETMARKERINFO command will return specified information about a specified Marker, from a specified sweep. In a file with multiple markers, the number of the marker is generally determined by the order that the marker was placed - not in the order that they appear on the x-axis (see the example below). The program will return "Max" even if you had used Latency, Amplitude, or Both. Therefore, it is a good idea to verify the label of the marker you select to be sure it is the one you intend. With AVG files, the 4th parameter is ignored.

1	string	Channel label
2	int	Marker index (zero-based)
3	defined value	parameter (XVALUE, YVALUE, LABEL)
4	int	Epoch index (ignored for AVG files)

**Return.** The information that is returned depends upon what you request.

**GETNUMCHANS** - Get Number of Channels.  This command will return the number of channels in any type of data file.


**Return.** The number of channels is returned.

**GETNUMDISPLAYPAGES** - Get Number of Display Pages.  This command will return the number of Display Pages that have been set up and are available to be selected by SETDISPLAYPAGE in any type of data file.

**Return.** The number of display pages is returned.


**GETNUMPOINTS** - Get Number of Points.  This command will return the number of data points in a sweep or file (CNT, AVG or EEG files).

**Return.** The number of points is returned.

**GETNUMSWEEPS** - Get the Number of Accepted/Rejected Sweeps.  This command will return the number of accepted or rejected sweeps in an AVG or EEG file.


**Return.** The number of accepted and/or rejected sweeps is returned.

1	defined value	Parameter (-Accepted, -Rejected)
---	---------------	----------------------------------

**GETOUTPUTFILE** - Get Output File.  This commands lets you create and display the standard Save As utility window. See also **GETINPUTFILE**.


1	string	Title
2(optional)	string	Default extension
3(optional)	string	Default filename
4(optional)	string	Initial path

**Return.** The full path of the file selected by the user.


**GETPOINTDATA** - Get Data Points.  These commands will return the actual data points in the specified range and sweep number (if EEG file). The commands are used with EEG and AVG files only.

**Return.** A list containing the data point values within the specified range and epoch are returned.


1	int	Start point (zero-based index)
2	int	End point (zero-based index)
3	string	Electrode label
4 (EEG files only)	int	Epoch number (zero-based index)

**GETSWDISPLAYSPEED** Get Display Speed.  These commands are used to return the current number of seconds in the display (the display speed; CNT files only). See also **SETSWDISPLAYSPEED**.


**Return.** The number of seconds displayed is returned.

**GETSWEEPMAX** - Get Sweep Stop Latency.  This command will return the stop latency (in ms) of the sweep of the current working file.

**Return.** The sweep maximum is returned. (in ms)


**GETSWEEPMIN** - Get Sweep Start Latency.  This command will return the start latency (in ms) of the sweep of the current working file.

**Return.** The sweep minimum is returned. (in ms)


**GFP** - Global Field Power.  The GFP command will compute the global field power waveform and compute a common average reference for all electrodes (refer to the EDIT manual for details). Use with CNT, EEG or AVG files. Use the **REFER** command to select a subset of channels to use as the reference.

1	Boolean	Compute GFP channel
---	---------	---------------------

2	Boolean	Include reference electrode
3	Boolean	Compute Z scores
4	string	Output file name

**GROUPAVG** - Group File Average.  This command is used to combine AVG files to form a group averaged AVG file. The formerly used **GROUP**, **INDIVIDUAL**, and **MINSWEEPS** commands have been discontinued. **Important: unlike almost all other data manipulation commands, this one does not use the current working file. In fact, there does not have to be any current working file for this transform to succeed.**


1	list	File names to be averaged (at least 2 files; "" will fail)
2	defined value	Grouping method (GROUP, INDIVIDUAL)
3	Boolean	Compute variance
4	int	Minimum sweeps
5	Boolean	Exclude Bad channels
6	Boolean	Exclude Skipped channels
7	string	Output file

**HERTZTOPOINT** - **NEW** Point Number for Hz.  This command is used to return the nearest point index (zero-based) to the Hz value you select (see the example below). "Point" in this case the frequency bin number. The command is used with frequency domain AVG and EEG files only. See also POINTTOHERTZ.


**Return.** The zero-based, frequency bin number nearest the selected Hz value is returned.

**HIGHIMPEDANCE** - High Impedances Mark Bad or Hide.  This command will automatically mark as Bad or Hide any channels that have impedances beyond a threshold you set.

1	int	Impedance limit (kOhms)
2	defined value	Mark as (-Bad, -Hide)
3	Boolean	Reset good electrodes


**IMPORTAVG** - Import AVG File from ASCII.  This command allows you to import average waveform/spectrum data that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the AVG type. If the information listed in the parameters below is already contained in the ASCII file header, then the parameters will be ignored. If the information is not in the header, then supply it using the parameters. You may wish to use the **READPOS** command (created from an existing, matching AVG file using the Overall Layout screen) to position the electrodes automatically (otherwise they will be displayed in a grid format). See also **EXPORTAVG**.

1	string	File name
2	defined value	Method (POINTS, ELECTRODES)
3	integer	Number of channels
4	double	Acquisition rate
5	double	X min
6	int	Number of points
7	int	Number of sweeps used to create the average
8	Boolean	Frequency domain


**IMPORTCNT** - Import CNT File from ASCII.  This command allows you to import continuous data that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the CNT type. If the information listed in the parameters below is already contained in the ASCII file header, then the parameters will be ignored. If the information is not in the header, then supply


it using the parameters. The last parameter is used to specify Neuroscan or Nihon Kodan data files (files are assumed to be Neuroscan CNT files by default - you may leave the 4th parameter blank if you are importing Neuroscan CNT files).

1	string	File name
2	int	Number of channels
3	double	Acquisition rate
4	defined value	Company (NEUROSCAN, NK)


**IMPORTEEG** - Import AVG File from ASCII.  This command allows you to import average waveform/spectrum data that have been saved in ASCII format. This command will read an ASCII file and create a binary file of the AVG type. If the information listed in the parameters below is already contained in the ASCII file header, then the parameters will be ignored. If the information is not in the header, then supply it using the parameters. You may wish to use the READPOS command (created from an existing, matching AVG file using the Overall Layout screen) to position the electrodes automatically (otherwise they will be displayed in a grid format). See also EXPORTAVG.

1	string	File name
2	defined value	Method (POINTS, ELECTRODES)
3	int	Number of channels
4	double	Acquisition rate
5	double	x Min
6	int	Number of points
7	int	Number of sweeps
8	Boolean	Frequency domain


**INSERT** - Insert.  The INSERT command allows you to add characters to a file name. When this command appears in the BATCH file, a prompt will appear on the screen requesting you to enter a character string. To insert the string use the % sign in the desired filename. This command can be used to automate a BATCH file in which the same file name will be used in several places throughout the file. Here is an example. Let's say you have several files named XXXfile, where XXX is a number. Select a specific file by entering 123 in the INSERT display, for example, to process the 123file.eeg. This command is included to provide backward compatibility with previous versions of BATCH. New commands such as GETINPUT, GETINPUTFILE and GETOUTPUTFILE should be used instead of INSERT when creating new BATCH files.

**INSERTSTIMEVENT** - Insert Stimulus Event.  This command allows you to insert a stimulus event at a specified point in a CNT file. To insert "Accept" and "Reject" type of events, use the REJECTBLOCK command.

1	int	Offset (number of points into the data file)
2	int	Stim code
3	int	Response code
4	double	Response latency
5	defined value	Accuracy (NORESPONSE, INCORRECT, CORRECT)


**INSERTRESPONSEEVENT** - **NEW** Insert Response Event.  This command allows you to insert a response event at a specified point in a CNT file. To insert "Accept" and "Reject" type of events, use the REJECTBLOCK command.

1	int	Offset (number of points into the data file)
2	int	Keypad code


**INSTRUCT** - User Instruction.  The INSTRUCT command allows the user to insert messages to be displayed on the screen while the program is in the command mode. These can be used as a prompt to the user. The command may also be used to return requested information.

1	string	Instruction
2(optional default =OK)	defined value	Type (OK, OKCANCEL, RETRYCANCEL, YESNO, YESNOCANCEL)


**Return.** One of the following: (depending on the type)  
 OK  
 CANCEL  
 RETRY  
 YES  
 NO

**INTEGRATEWAVEFORM** - Integrate Waveform.  This command performs an accumulating sum across each sweep for each channel (refer to the EDIT manual for more details).

1	Boolean	Rectify
2	string	Output file (required for AVG and EEG files)

**INTRACLASSCORR** - Intra-class Correlation.  The intra-class correlation statistic is a measure of overlap and related variability between two waveforms (from two AVG files). Retrieve one file, then enter the BATCH command on the next line.


1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.dat)

**INTRACLASSCORR\_EX** - **NEW** Intra-class Correlation, extended.  This command differs from the INTRACLASSCORR command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").


1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.dat) (or "")
5	Boolean	Display results
6	list	Channels to include

**INVERSEFFT** - Inverse FFT.  The INVERSEFFT command restores data to time domain for EEG files that have been transformed using the FORWARDFFT transform. See also FORWARDFFT.

1	string	Output file name
---	--------	------------------


**ISOFFLINE / ISONLINE** - **NEW** Return Online or Offline.  These commands are used primarily in conjunction with the autorun feature. You may want to have a different set of functions loaded depending on whether you are online (in ACQUIRE) or offline (in EDIT). The commands will return either a 0 or 1, depending on which program you are in (0 means no, and 1 means yes). No parameters are used with either command.

**Return.** Either a 0 or a 1 is returned.


**LATENCYTOPOINT** - Latency to Point Index.  This command is used to return the index number of the data point that corresponds to a given latency (AVG and EEG files only). For example, the GETPOINTDATA command is used to return the value of a given data point, or set of points. It requires the index number of the data points (not the millisecond values). Knowing the AD rate and the start time of the sweep, you can calculate the index numbers, or, you can use LATENCYTOPOINT. Enter the latency of a data point, and the command will return the index number of the point. See also POINTTOLATENCY.

**Return.** The point number (0 based index) nearest the latency is returned.


1	double	Latency (in ms)
---	--------	-----------------

**LDR** - Linear Derivation.  The Linear Derivation transform enables the creation of new channels as arbitrary linear combinations of existing channels (see the EDIT manual for more details).


1	string	LDR file name
2	string	Output file name
3 (opt)	string	Units (default used if not specified)

**LOAD3DD** - **NEW** Load 3DD File.  This command is used to load a 3DD electrode position file (from 3DSpaceDx) to whichever data file has the focus.


1	string	3DD file name (or "")
---	--------	-----------------------

**MAXIMIZE** - **NEW** Maximize Window.  This command performs the standard Windows Maximize function. Applying it will maximize the display of the window that has the focus. Use SELECTFILE to change the focus to the desired window, as needed.


*Example.* MAXIMIZE. No additional parameters are used.

**MINIMIZE** - **NEW** Minimize Window.  This command performs the standard Windows Minimize function. Applying it will minimize the display of the window that has the focus. Use SELECTFILE to change the focus to the desired window, as needed.


*Example.* MINIMIZE. No additional parameters are used.

**MERGE** - Merge.  This command is used to append one or more EEG files to a previously opened file. The command is applied only to epoched (EEG) files.


1	list	Files to merge
2	string	Output file name

**MERGEevt** - Merge Event File.  This command is used to merge behavioral data (the DAT file from STIM) with its matching continuous data file. The command is used only with CNT files. The MERGEevt command replaces the discontinued TASKFILE command.


1	string	Task file name
---	--------	----------------

**NEWNAME** - New File Name.  This command may be inserted at any point in the batch file, and is used to rename a file.


1	string	File name ("" will fail)
2	string	New file name ("" will fail)

**OPENFILE** - Open File.  This command is used to open any types of data file used in the EDIT program. The last opened file will be the current working file. See also CLOSEALL, CLOSEFILE, and SELECTFILE.


1	string	File name
---	--------	-----------

**PAIRED** - Paired t-test.  A paired t-test is used when comparing two related or matched groups, or when comparing test-retest measures on the same group. See the EDIT manual for more details.


1	string	Output file name
---	--------	------------------

**PAUSE** - Pause.  The PAUSE command will halt execution of a BATCH file until the Resume button is pressed. This command is useful to halt a BATCH command at a specific point so that the user can perform some type of interaction. It can be used with an option time to wait. This is useful to delay the execution of the next command for a specified period of time.


1 (optional)	int	Time to wait (in ms's)
--------------	-----	------------------------

**PAUSEDISPLAY** - Pause Display.  This command is used to pause or un-pause the display without interrupting data storage.

1	Boolean	Pause display
---	---------	---------------



**PEAKDETECTION** - Peak Detection.  This command performs an automatic peak detection placing markers at those locations.

1	string	Marker label
2	Boolean	User defined interval
3	int	Start search point (ignored if param 2 is FALSE)
4	int	Stop search point (ignored if param 2 if FALSE)
5	defined value	Search type (NONE, MAX, MIN)
6	Boolean	Locked
7	string	Master channel label (ignored if param 6 is FALSE)
8	Boolean	Exclude bad channels
9	Boolean	Exclude skipped channels
10(opt)	defined value	Label type (USER, LATENCY, AMPLITUDE, BOTH)



**PEAKDETECTION\_EX** - **NEW** Peak Detection, Extended.  This command differs from the PEAKDETECTION command in that it allows you to list the electrodes that you want to include for peak detection (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").

1	string	Marker Label
2	Boolean	User Defined interval
3	float	Start search latency (ignored if param 2 if false)






4	float	Stop search latency (ignored if param 2 if false)
5	defined value	Search Type (NONE, MAX, MIN)
6	Boolean	Locked
7	string	Master Channel Label (ignored if param 6 is false)
8	Boolean	Exclude Bad channels
9	Boolean	Exclude Skipped channels
10	defined value	Label type (USER, LATENCY, AMPLITUDE, BOTH)
11	defined value	Color (RED, ORANGE, YELLOW, GREEN, BLUE, LTBLUE, BLACK, WHITE)
12	list	Channels to mark (or "All")

**PEARSONS** - Pearson's r Correlation.   This operation computes Pearson's r correlation coefficients between paired electrodes within a specified latency range.

1	string	Reference file name
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.DAT)






**PEARSONS\_EX** - **NEW** Pearson's r Correlation, extended.   The PEARSONS\_EX command is the same as the PEARSONS command except for the new 5th parameter that lets you specify the channels to include.

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	string	Output file name (*.DAT) (or "")
5	Boolean	Show results
6	list	List of channels to include

**PICKOFFDIRECTORY** - Pick Off the Path and Folder.      The complete path and destination folder will be extracted from the string (includes all backslashes).






PICKOFFDIRECTORY c:\scan4.3\Batch Examples\viscpt.cnt yields c:\scan4.3\Batch Examples\.

1	string	File name and path
---	--------	--------------------

**PICKOFFEXTENSION** - Pick Off the File Extension.      The file extension, including the leading "dot", will be extracted.

PICKOFFEXTENSION c:\scan4.3\Batch Examples\viscpt.cnt yields .cnt

1	string	File name and path
---	--------	--------------------

**PICKOFFFILENAME** - Pick Off the Complete File Name.      The complete file name, including the extension will be extracted.

PICKOFFFILENAME c:\scan4.3\Batch Examples\viscpt.cnt yields viscpt.cnt


1	string	File name and path
---	--------	--------------------



**PICKOFFNAMEONLY** - Pick Off the File Name.  The file name, without the extension will be extracted.


PICKOFFFILENAMEONLY c:\scan4.3\Batch Examples\viscpt.cnt yields viscpt.

1	string	File name and path
---	--------	--------------------

**POINTTOHERTZ** - **NEW** Hz for Point Number.  This command is used to return the nearest Hz frequency to the zero-based bin number value you select (see the example below). "Point" in this case the frequency bin number. The command is used with frequency domain AVG and EEG files only. See also HERTZTOPOINT.

**Return.** The Hz value nearest the selected zero-based, frequency bin number is returned.


1	int	Point (frequency bin) value
---	-----	-----------------------------


**POINTTOLATENCY** - Point Index to Latency.  This command is the complement to the LATENCYTOPOINT command, and is used to return the latency nearest to the data point you specify (EEG and AVG files only). Enter the zero-based index of the data point, and the command will return the latency (in ms) of the data point.

**Return.** The latency of the data point is returned (in ms)


1	int	Index of the data point (zero-based index)
---	-----	--

**PRINT** - Print.  This command will print the contents of the currently active window.


**PRINTWAVEBOARD** - **NEW** Print Waveboard Display.  This command is used to Print the display in the Waveboard (no additional parameters are used).

**READPOS** - Read Electrode Position Files.  The READPOS command will read an ASC file, and position the electrodes on the screen according to the new coordinates. See also SAVEPOS.



1	string	Position file name (ASC extension)
---	--------	------------------------------------

**READREJ** - Read Rejected Sweeps File.  The READREJ command reads a REJ file and rejects individual sweeps specified in the file. REJ files are created with the SAVEREJ command. This command can be used to recall a list of sweeps that have been previously rejected. For example, an EEG file could be visually edited for artifact, the list of rejected sweeps saved with the SAVEREJ command, and then recalled later with the READREJ command to reapply the manual rejection criteria. (The \*.rej file is an ASCII string of 0's and 1's, where 0=reject, and 1=accept).



1	string	Rejected sweeps file name
---	--------	---------------------------

**READSUB** - Read Subject Information File.  The READSUB command will load a subject information file. Subject information files have a .SUB extension, and may be saved manually from ACQUIRE, or with the SAVESUB command.



1	string	Subject file name
---	--------	-------------------

**RECTIFY** - Rectify File.   The Rectify transform is a simple “absolute value” operation: all positive waveform values are left unchanged, and all negative waveform values are inverted to their corresponding positive values. It may be used with EEG and AVG file types. The output file is required for EEG files and is omitted for AVG files.



1(only for EEG files)	string	Output file name
-----------------------	--------	------------------

**RECTIFY\_EX** - **NEW** Rectify File, extended.   The RECTIFY\_EX command differs from the RECTIFY command only in that you may select the channels to include (new parameter 1).



1	list	List of channels to include (or "ALL")
2(optional)	string	Output file name (or "")



**REFER** - **NEW** Create New Reference.   The REFER command will compute a new reference using the electrodes designated as reference sites. (Note: this command exists in BATCH only. In point and click mode, use the GFP/Reference transform, and select the desired channels to be included in the new reference).

1	Boolean	Compute GFP
2	Boolean	Include reference electrode
3	Boolean	Compute field z-scores
4	list	Channels to include (or "All")
5	string	Output file




**REJECTBLOCK** - Reject Block.   Use this command to reject blocks of data in CNT files. (You will need the AD rate to compute the offset:  $\text{offset} = \text{AD}/1000 \times \text{ms}$ ).

1	int	Start offset
2	int	Stop offset



**REM** - Remark.   The REM command is used to place text in the BATCH file that is not recognized by the program. It is used interchangeably with the # sign.

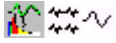
**REMOVEDISPLAYFILTER** - Remove the Display Filter.   This command is used to remove display filters to the active data file. See also ADDDISPLAYFILTER.


1	list	Channels to remove display filter (or “All”)
---	------	--


**RENAME** - Rename Electrode.    This option is used to rename *electrode labels*. Retrieve the data file, and enter the old name and the new name for the label to be renamed. Save the file with the new labels, if desired.


1	string	Electrode name
2	string	New name (maximum 19 characters)

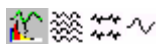
**RESETFORBASECOR** - Reset All Channels for Baseline Correction.   This command will reset all channels for baseline correction. It is typically used where some channels had been excluded for baseline correction. It avoids having to reset each channel individually. See also EXCLUDEFORBASECOR and BASECOR.


**RESETFORDETREND** - Reset All Channels for Linear Detrend.  This command will reset all channels for linear detrending (so that none are excluded). It is typically used where some channels had been excluded for detrending. It avoids having to reset each channel individually. See also DETREND and EXCLUDEFORDETREND.

**RESETFORFILTER** - Reset All Channels for Filtering.  This command will reset all channels for filtering. It is typically used where some channels had been excluded for filtering. It avoids having to reset each channel individually. See also FILTER and EXCLUDEFORFILTER.


**RESETFORPEAKDETECTION** - Reset All Channels for Peak Detection.  This command will reset all channels for peak detection. It is typically used where some channels had been excluded for peak detection. It avoids having to reset each channel individually. See also PEAKDETECTION and EXCLUDEFORPEAKDETECTION.

**RESETFORSMOOTH** - Reset All Channels for Smoothing.  This command will reset all channels for smoothing. It is typically used where some channels had been excluded for smoothing. It avoids having to reset each channel individually. See also SMOOTH and EXCLUDEFORSMOOTH.

**RESETFORWINDOWDATA** - Reset All Channels for Windowing.  This command will reset all channels for windowing. It is typically used where only some channels had been set for windowing. It avoids having to reset each channel individually.


**RESPWIN** - **NEW** Response Window.  The RespWin (Response Window) program performs an offline analysis of behavioral data in an event (.EVT) file that consists of a stream of stimulus and response events. The user defines a response window and one or more paired subsets of stimuli that correspond to target events and associated distractor events. A hierarchy of rules is applied to resolve ambiguities in forming associations between stimuli and responses. Hits, correct rejections, false alarms, and misses are tallied, and a nonparametric signal detection analysis is given. Reaction times are computed for each stimulus-response association (see the EDIT manual for details). Repeat the last 5 parameters for each target/distractor pair (within the same RESPWIN command).

1	double	Window minimum (ms)
2	double	Window maximum (ms)
3	Boolean	No Intervening Stimuli
4	string	Output event file name (.ev2 extension)
5	string	Output summary file name (.sum extension)
6	Boolean	Include header in the event file
x	string	Descriptor
x	string	Target vector (1, 2, 3-6, etc.)
x	int	Target correct response
x	string	Distractor vector (1, 2, 3-6, etc.)
x	int	Distractor correct response (or 0)


**RESTART** - Restart Acquisition.  This command serves the same purpose as the Restart icon on the ACQUIRE Toolbar, is used to restart the acquisition process.

**RESTORE** - **NEW** Restore Window.  This command performs the standard Windows Restore function. Applying it will Restore the display of the window that has the focus to its previous size. Use SELECTFILE to change the focus to the desired window, as needed.


*Example.* RESTORE. No additional parameters are used.

**REVIEW** - Review Sweeps.  The REVIEW command has the same function as the PAUSE command - it suspends the BATCH file at the point where the command is inserted. In EDIT, this allows you to, for example, review and edit epoched files for artifact in the interactive single-sweep mode. When used with the "Time to wait" argument, the selected file(s) will be displayed for the designated time span. To terminate the REVIEW command, press the Resume button on the Batch dialog display. All menu items and Toolbar icons in EDIT are active while the REVIEW command is in effect.


1 (optional)	int	Time to wait (in milliseconds)
--------------	-----	--------------------------------

**SAVEAS** - Save File As.  The SAVEAS command is used to save data files. You cannot overwrite an EEG or CNT file that is currently open (use a different name). See also OPENFILE.


1	string	Output file name
---	--------	------------------

**SAVEPEAK** - Save Peak File.  The SAVEPEAK command saves the peak detection information to the currently open peak file. If the -SubstituteBad flag is specified then the next parameter must be a string that will be placed in the location of peak data for all bad electrodes. This is a convenient way of flagging additional analysis that these electrodes were bad.


1	string	File name (or "")
2 (optional)	defined value	option (-SubstituteBad)
3	string	Substitution string

**SAVEPOS** - Save Electrode Position Files.  The SAVEPOS command will save an ASC file specified by the filename argument that contains the positions of the current electrode displays. See also READPOS.


1	string	Position file name (ASC extension)
---	--------	------------------------------------

**SAVEREJ** - Save Rejected Sweeps File.  The SAVEREJ command saves a REJ file to disk containing a table of sweeps that have been rejected in the EEG file. REJ files are read with the READREJ command.



1	string	Rejected sweep file name
---	--------	--------------------------

**SAVESUB** - Save Subject Information File.  The SAVESUB command will save a subject information screen to disk. Subject information files have a .SUB extension and can be read manually from ACQUIRE or EDIT, or in BATCH with the READSUB command. This can be useful if, for example, a protocol call for serial recordings on multiple days for the same subject. See also READSUB.



1	string	Subject file name (*.sub)
---	--------	---------------------------

**SAVEEVENT** - Save Event File.  The SAVEEVENT command performs a SCAN of a continuous (CNT) file and writes an event file (EVT or EV2) to disk containing the file position and tag of each event. If a task file is loaded prior to the SAVEEVENT command, behavioral data (accuracy, reaction time, responses and stimuli) will be included in the event file. In versions prior to 3.0 the event file can be recalled prior to the EPOCH command to increase the speed of flexibility of event windowing. This is not necessary in 3.0 versions and later since the event information is stored in a table for rapid lookup during epoch creation. With 3.0 and later versions, the MERGEEVT command is recommended.



1	Boolean	Include Stimulus events
2	Boolean	Include KeyPad events
3	Boolean	Include Keyboard events
4	Boolean	Include Rejected sweeps
5	Boolean	Use seconds
6	Boolean	Include header
7	string	Output event file name (*.evt= old style, *.ev2= new style; or "")

**SAVEEVENT\_EX** - **NEW** Save Event File, Extended.   The **SAVEEVENT\_EX** command adds the additional options to save the DC Correction and Start/Stop events to the event file.


1	Boolean	Include Stimulus events
2	Boolean	Include KeyPad events
3	Boolean	Include Keyboard events
4	Boolean	Include Rejected sweeps
5	Boolean	Use seconds
6	Boolean	Include header
7	Boolean	Include DC corrections
8	Boolean	Include Start/Stop events
9	string	Output event file name (*.evt= old style, *.ev2= new style; or "")

**SCALE** - Scale the Displayed Data.   The **SCALE** sets the min/max scale range for screen display (Multiple Windows files only), and will not affect the data. This command is useful in setting a series of files to the same display scale range.



1	double	Minimum voltage
2	double	Maximum voltage

**SCALEX** - **NEW** Scale the X-axis.   The **SCALEX** command sets the min/max time scale range for the screen display (EEG and AVG files only). If you save the modified file, the changes will be permanent; you cannot go back to the original start/stop times.





1	int	Minimum time point
2	int	Maximum time point

**SELECTFILE** - Select Open Data File.  This command allows you to switch the current working file to one of several data files that you may have open. Once selected, you may perform further processing of the selected file, while leaving the other files open. The **BATCH** processor directs all data manipulation commands to the current working file. The current working file will remain in effect until another file is opened, another file is selected, or the current file is closed. See also **OPENFILE**, **CLOSEFILE**, and **CLOSEALL**.



1	string	Data file to be selected (" " will fail)
---	--------	--

**SETBASELINECOR** - Set Baseline Display Correction.   Used to apply or remove the correction. This command has the same function as clicking the Baseline Correction icon from the Toolbar (it affects the display only - not the stored data), and is used with **CNT** files only. See also **GETBASELINECOR**.



1	Boolean	Enable Baseline Correction
---	---------	----------------------------

**SETCHANATTRIBUTE** - Set Channel Attribute.     This command is used to set the selected channel attribute, as desired (use with any file type). The command can be used with single, several, or all channels in the file. See also GETCHANATTRIBUTE.

1	list	Channel label(s) or “All”
2	defined value	Attribute Type (-Artifact, -Fsp, -Hide, -Skip, -Bad, -AutoAdd, -AutoAddLast)
3	Boolean	Set attribute

**SETCOHREF** -**NEW** Set the Coherence Reference.   (COH files only) This command is used to set or change the Coherence Reference.

1	list	Electrode label
---	------	-----------------

**SETCURRENTEPOCHINDEX** - Set Current Epoch Index.   This command will set the currently displayed epoch of an EEG file to the desired epoch. Set also GETCURRENTEPOCHINDEX and GETEPOCHCOUNT.



1	int	Sweep number of the epoch to display (zero-based index)
---	-----	---

**SETDISPLAYINVERTED** - Set Display Inversion.     This command is used to set the display polarity. See also GETDISPLAYINVERTED.

1	Boolean	Inverted
---	---------	----------

**SETDISPLAYPAGE** - Set Display Page.     This command is used to display a specified display page. See also GETNUMDISPLAYPAGES and GETDISPLAYPAGE.

1	int	Display page index (0 based)
---	-----	------------------------------

**SETEPOCHINFO** - Set Epoch Information.   This command will make changes for specified epochs. See also GETEPOCHINFO.

1	int	Epoch number (zero-based index)
2	defined value	Parameter (-TrialType, -Accept, -Correct, ReactionTime, -Response)
3	varies	New value to replace the existing one
-TrialType	T set the stimulus type code number	
-Accept	0 = Reject, 1 = Accept	
-Correct	0 = Incorrect, 1 = Correct	
-ReactionTime	To set the latency of the response	
-Response	To set the response type code number	

**SETEVENTINFO** - Set Event Information.   This command is used to modify information in the event table of CNT files. See also GETEVENTINFO.


1	int	Event number (zero-based index)
2	defined value	Parameter (-Offset, -ResponseLatency, -Accuracy, -KeyboardCode, -KeypadCode, -StimulusCode)

3 varies Based on list below:

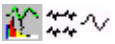
-Offset	int	Number of points since the beginning of the file.
-ResponseLatency	double	The response latency in ms.
-Accuracy	defined value	NORESPONSE, INCORRECT, CORRECT
-KeyboardCode	int	Function key number (2-11)
-KeypadCode	int	Number of response pad button pressed (1-4)
-StimulusCode	int	The stimulus type code number (1-255)

**SETSPECTDISPLAYSTYLE** - **NEW** Set Spectral Display Style.  This command is used to select the style of the spectral display - line or histogram. It is used with frequency domain AVG and EEG files.


1 defined value Style (LINE, HISTOGRAM)

**SETSWDISPLAYSPEED** - Set Display Speed.  This command is used to set the number of seconds in the display (CNT files only). See also GETSWDISPLAYSPEED.


1 int Seconds in display (2-20 are valid entries)

**SMOOTH** -Smooth.  The SMOOTH command is used to average each data point with adjacent points to create a smoother waveform. The command may be used with EEG or AVG files. The smoothed results are written to a EEG new file. Smoothing with AVG files is done "in place" - you will need to save the smoothed AVG file with the SAVEAS command, if desired. See also EXCUDEFORSMOOTH, RESETFORSMOOTH and SMOOTH\_EX.

1	int	Number points in smoothing (must be an odd number, at least 3)
2	int	Number of passes through the data
3(optional)	string	EEG output file name (omitted for AVG files)

**SMOOTH\_EX** -**NEW** Smooth, extended.  This command differs from the SMOOTH command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX** Commands" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All"). The smoothed results are written to a EEG new file. Smoothing with AVG files is done "in place" - you will need to save the smoothed AVG file with the SAVEAS command, if desired.


1	int	Number of points in smooth (must be an odd number, at least 3)
2	int	Number of passes through data
3	list	List of electrodes to smooth (or "All")
4	string	Output file name

**SNR** - **NEW** Signal to Noise Ratio.  (time domain only) The Signal to Noise Ratio (SNR) transform computes "noise" over a specified interval, "signal" over a specified interval, and then the ratio of signal over noise. "Signal" and "noise" are defined as the variance among data points in the selected interval. The SNR then is the ratio of the variance (signal) over the variance (noise). You can specify the interval to use for the noise and signal. Typically, you would use the pre-stimulus interval to estimate the noise (or the 20th percentile if there is no prestim interval), and some or all of the post-stimulus interval to estimate the signal. You also have the options to use the Entire Sweep for the Signal, or a User Defined interval for either or both.


1	defined value	Interval (PRESTIMINTERVAL, PERCENTILE, USERDEFINED)
2	double	Interval Noise start latency




3	double	Interval Noise stop latency
4	defined value	Interval (POSTSTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
5	double	Interval Signal start latency
6	double	Interval Signal stop latency

**Sort** - Sort Sweeps.  The Sort command will create a new EEG file that contains only the sweeps that meet the specified (used with EEG files only). For latency and accuracy information, you will need to have merged the DAT file from STIM (MERGEEVT) with the CNT file prior to epoching (EPOCH) the CNT file. The SORT command is used to apply the sorting criteria you create using the CREATSORT command.

1	string	Sort name
2	string	Output file name

**SPATIALFILTER** - Spatial Filter.  The spatial filter transform removes and/or retains signals in spatial subspaces of the full measurement space in the context of a control signal (refer to the EDIT manual for more details). It may be used with CNT, EEG or AVG files. To use the command, you will need to have already created the Removal, Retention, and/or Control LDR files (see the VEOG correction examples in the Spatial Filter section of the EDIT manual for ways to create these). Different reconstitution options are available depending upon which type(s) of LDR files you are using (Removal, retention, etc.). If you are not completely familiar with using the Spatial Filter, you should set up the procedure in point&click mode first, then transfer the settings manually to the Tcl command line.

1	string	Removal LDR
2	string	Retention LDR
3	string	Control LDR
4	string	Output spatial filter LDR file
*****Reconstitution of Original Channels*****		
5	Boolean	Reconstitute original channels
6	Boolean	Remove unwanted signals
7	Boolean	SNR Transform
8	Boolean	Project signals of interest only
9	Boolean	Include skipped channels
*****Derivation of Signal channels*****		
10	Boolean	Derive channels for unwanted signals
11	Boolean	Derive channels for Signals of interest
12	string	Output file name (CNT files only)

**SPATIALSVD** - Spatial SVD.  The spatial singular value decomposition (SVD) transform generates spatial component topographies, fraction of total variance explained by each component, and a linear derivation file for deriving component time series. This transform can be applied to any time domain file type (AVG, EEG, or CNT). Please refer to the EDIT manual for more details. See also CREATSORT.


1	defined value	Covariance/Cross product (COVARIANCE, CROSSPRODUCT)
2	Boolean	Include skipped
3	double	Start latency
4	double	Stop latency
5	double	Percent retained variance (ignored if DATAMATRIX is selected)
6	string	Output LDR file
7	defined value	Output LDR type (COMPONENTS, DATAMATRIX)




8	string	Sort name
9	string	Output AVG file


**SPECTRUM** - Spectrum.  The SPECTRUM command will compute a Fast Fourier Transform on an AVG file.

1	defined value	Spectrum type (MAGNITUDE, PHASE)
2	defined value	Scaling method (ignored if param 1 is PHASE) (AMPLITUDE, POWER)
3	defined value	Window type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
4	double	Window taper percent
5	string	Output file name

**SPLINEFIT** - Spline Fit.  The SPLINEFIT command will perform a spline interpolation of the current waveform average. The command can be used to fit an arbitrary number of points to the waveform, but it is particularly useful when large numbers of waveform averages have been created with a non-power-of-two number of points, but the user desires to compute FFT spectra on these waveforms. When epoched or continuous data is acquired under a set of parameters that leads to waveform averages with a non-power-of-two number of points, the number of points in the waveform average can be changed with the SPLINEFIT command. The SPLINEFIT command can also be used to change the number of points in one set of waveform averages so that they can be group averaged with another set of waveforms which were created with a different number of points. The command is used with EEG and AVG files.


1	int	Points
2	string	Output file name


**STARTACQUISITION** - Start Acquisition.  Use this command to initiate viewing of the incoming signals. (It has the same functions as the older VIEW command). You may want to place a PAUSE command after it. This will let the incoming signals scroll until you want to begin data storage (then click the "Resume" button).


**STARTRECORDING** - Start Recording.  Use this command to initiate data storage. If a file name is specified then the data will be stored to the specified file. It is important to use the correct extension for the type of data being stored. If no file name is specified, a Save As screen will appear allowing you to enter a path and file name. You may want to place a PAUSE command after it. This will let data storage continue until you want to stop it (then click the "Resume" button).

1(optional)	string	file name
-------------	--------	-----------

**STOPACQUISITION** - Stop Acquisition.  Use this command to close the acquisition display.

**STOPRECORDING** - Stop Recording.  Use this command to end data storage.

**SUBJECT** - Subject Information Display.  The SUBJECT command will display the subject information screen, in which you may enter subject information. This command can be useful in automating the acquisition process. Save the subject information form the Subject display, or with the SAVESUB command. See also SAVESUB and READSUB.


**SUBTRACT** - Subtract Files.  The SUBTRACT command is used to perform a point by point mathematical subtraction of the selected AVG file to the one specified by the first parameter and create a

new output file. This is used, for example, to compute the difference waveform between the rare and common responses of a p300 recording.



1	string	File name to subtract
2	string	Output file name


**TOGGLEZOOM** - Toggle Zoom.  This command is used to toggle between zoomed in and zoomed out electrode displays. See also ZOOMIN and ZOOMOUT.


1	string	Electrode name
---	--------	----------------

**TSCORE** - t-test.  The TSCORE command calculates t-test comparisons between two waveform files which have the optional variance waveforms as well as the normally displayed mean-value waveforms. For example, this option would typically be used for the comparison data from two independent groups. Retrieve one group averaged file, then use the TSCORE command.


1	string	Reference file name
2	string	Output file name

**UPDATECHANGES** - **NEW** Update Changes to Data file.  The UPDATECHANGES command is used to save changes to the open data file that has the focus. It has the same function as the  icon on the Toolbar. The command will overwrite the open file, saving the changes with the file when you close it. The command exists on a line by itself, and uses no parameters. For example, say retrieve a file and perform peak detection in BATCH, and you wish to save the changes while leaving the file open. The UPDATECHANGES will update the open file, saving the changes when you close the file. The command is not used with COH files.

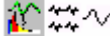
**VIEW** - View Incoming Signals.  Use this command to initiate viewing of the incoming signals. (It has the same functions as the newer STARTACQUISITION command which should be used instead). See also STARTACQUISITION.

**VOLTAGETHRESHOLD** - Voltage Threshold.  The Voltage Threshold option allows you to insert event markers into a CNT file (only) on the basis of a detected voltage in a specified channel, and to reject sections of a CNT where detected voltages exceed a specified threshold.


1	defined value	Operation type (INSERTEVENTS, REJECTSEGMENTS)
2	string	Trigger Channel
3	defined value	Threshold Type (GREATERTHAN, LESSTHAN, ABSOLUTEVALUE)
4	int	Stim code (ignored if operation is not INSERTEVENTS)
5	double	Refractory period
6	double	Threshold

**WINDOWDATA** - Window Data.  The Window transform applies a window taper to single-sweep epochs (EEG or AVG files). Application of a window can be useful to minimize edge effects. See also EXCLUDEFORWINDOWDATA and RESETFORWINDOWDATA.


1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Window percent
3(optional)	string	Output file name

**WINDOWDATA\_EX** - NEW Window Data, extended.  This command differs from the WINDOWDATA command in that it allows you to list the electrodes that you want to include (or "All"). See the "**\_EX Commands**" section above. The "Auto-write" feature will create an "\_EX" command automatically using the electrodes you specify in the "Select" dialog box (or "All").


1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Window percent
3	list	List of electrodes to include (or "All")
4(optional)	string	Output file name (or "")

**WRITELOG** - Write to Log File.  This command is used to specify the text string that is contained in the LOG file. The text string can be any information you want to save in a separate LOG file.


1	string	Log file name
2	string	Text to be included in the LOG file

**WRITENUMSWEEPS** - Write Number of Sweeps.  This command will create an ASCII output file containing the number of sweeps for each channel (it is typically used with averaged files, where you wish to see/store how many sweeps were included in each channel's average).


1	string	file name
---	--------	-----------

**ZOOMIN** - Zoom In.  The ZOOMIN command will zoom-in and display a waveform/spectrum at the site named in the electrode argument. This command is used with epoched or averaged data files. See also TOGGLEZOOM and ZOOMOUT.

1	string	Electrode name
---	--------	----------------

**ZOOMOUT** - Zoom Out.  The ZOOMOUT command will zoom-out and display a waveform/spectrum at the site named in the electrode argument. This command is used with epoched or averaged data files. See also TOGGLEZOOM and ZOOMOUT.

1	string	Electrode name
---	--------	----------------

**ZSCORE** - Z-score.  The ZSCORE command calculates Z-scores between two waveform files. This command would typically be used for statistical comparison of a waveform from an individual with an averaged waveform from a group.

1	string	Reference file name
2	string	Output file name