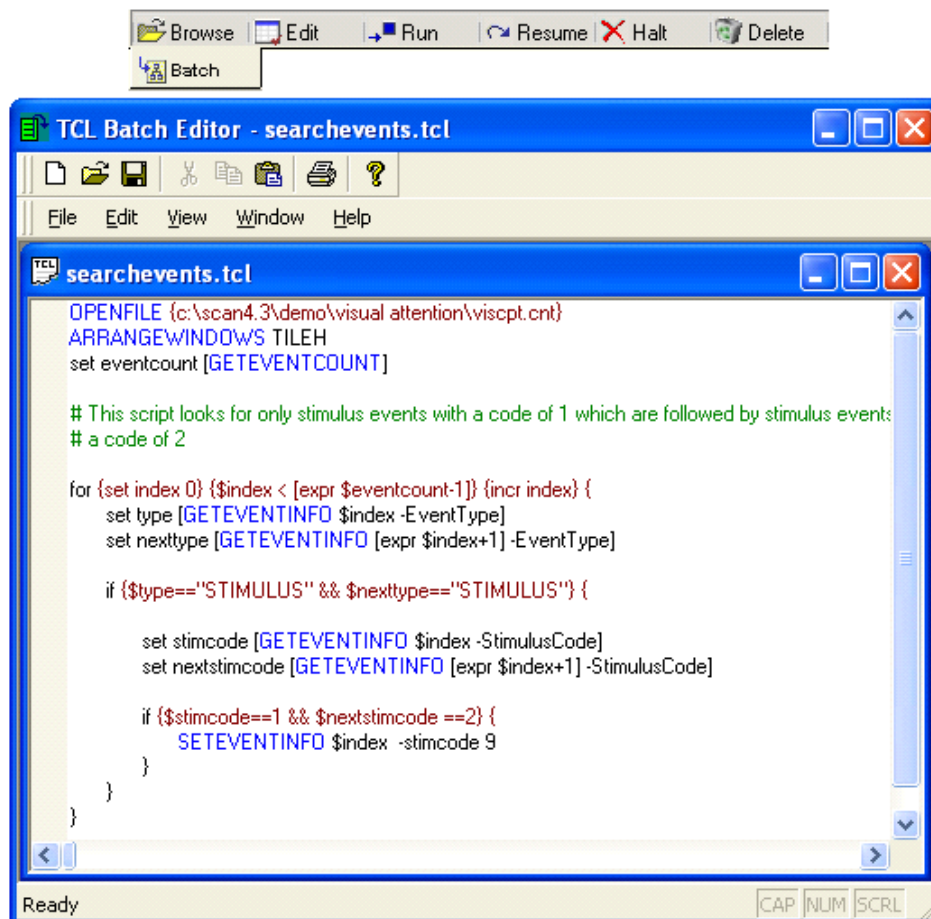


Tcl Batch Commands 4.5



Automated Data Analysis



Table of Contents

Part I Tcl Batch	1
1 Contact Information	2
Part II Introduction	3
1 Executing BATCH Files in SCAN	3
2 Tcl BATCH Command Structure	5
3 Tcl BATCH Command Conventions	6
4 Creating BATCH Files	10
5 Conditional and Looping Commands	12
6 Get and Set Commands	15
7 Variable Substitution	16
8 Using Autorun	21
9 Sample BATCH Files	23
10 Executing BATCH Files Outside of SCAN	24
Part III Tcl BATCH Commands	25
1 Tcl Commands for ACQUIRE and EDIT	26
ADDDISPLAYFILTER/REMOVEDISPLAYFILTER	26
ADDWAVEBOARD	26
ARRANGEWINDOWS	27
BUTTERFLYPLOT	28
CHANLIST	28
Data Scaling Commands	29
AUTOSCALE	29
SCALE	30
SCALEX	30
File Management Utility Commands	30
DELETE	30
NEWNAME	31
RENAME	31
GET/SET Commands	31
GETBASELINECOR/SETBASELINECOR	31
GETCHANATTRIBUTE	32
SETCHANATTRIBUTE	33
GETCHANLABEL	33
GETCHANNELINDEX	34
GETDATE/GETTIME	34
GETDISPLAYINVERTED/SETDISPLAYINVERTED	35
GETDISPLAYPAGE/SETDISPLAYPAGE	36
GETINPUT	36
GETINPUTFILE	37
GETNUMCHANS	38
GETNUMDISPLAYPAGES	38
GETNUMPOINTS	39
GETNUMSWEEPS	40

GETOUTPUTFILE	40
GETSWDISPLAYSPEED/SETSWDISPLAYSPEED	41
GETSWEEMAX	42
GETSWEEMIN	42
HERTZTOPOINT	42
INSTRUCT	43
ISOFFLINE/ISONLINE	44
LATENCYTOPOINT	44
MAXIMIZE	45
MINIMIZE	45
PAUSE	45
"PICKOFF" commands	46
POINTTOHERTZ	47
POINTLATENCY	47
PRINTWAVEBOARD	48
READPOS/SAVEPOS	48
REM	49
RESTORE	49
REVIEW	49
SETSPECTDISPLAYSTYLE	50
Subject Information Commands	50
READSUB	50
SAVESUB	50
SUBJECT	51
Substitutions	51
INSERT	51
WRITELOG	51
ZOOM Commands	52
TOGGLEZOOM	52
ZOOMIN	52
ZOOMOUT	52
2 Tcl Commands for ACQUIRE	53
CALIB	53
CLEARAVG	53
DODCCORRECT	53
DOIMPEDANCE	53
GETAST	54
HIGHIMPEDANCE	54
PAUSEDISPLAY	54
RESTART	55
STARTACQUISITION	55
STARTRECORDING	55
STOPACQUISITION	56
STOPRECORDING	56
Various Acquisition Script Examples	56
VIEW	58
Discontinued commands for ACQUIRE	58
3 Tcl commands for EDIT	58
ADD	60
ADDANNOTATION	60
Appending Data Files	60
APPEND	60
APPENDRECORDING	61

CONCATCNT	61
MERGE	61
AREAREPORT	62
Artifact Correction/Rejection	62
ARTCOR	63
ARTREJ	63
ARTREJ_EX	64
ARTREJCNT	64
ARTREJCNTFREQ	65
ARTREJCNTTRIG	65
ARTREJFREQ	65
ARTREJFREQ_EX	66
BLINKNOISEREDUCTION	66
CLEARART	67
EKGNOISEREDUCTION	67
FMRI	68
READREJ	69
SAVEREJ	70
Averaging	70
AVERAGE	70
AVERAGE_EX	70
GROUPAVG	71
AVGBANDS	71
Baseline Correction	72
BASECOR	72
BASECOR_EX	73
BASECOR_EX2	73
EXCLUDEFORBASECOR	73
RESETFORBASECOR	73
CLEARALLMARKERS	74
CLEARMARKERS	74
CLOSEALL	74
CLOSEFILE	74
COHERENCE	75
COHERENCE_EX	75
SETCOHREF	76
COMPAREELECTRODES	76
CORRELATESPEAKS	76
CREATESORT/DELETESORT	77
CUTEPOCH	78
DCCORRECT	78
DEBLOCK	79
DECIMATE	79
DELETEBADCHANNELS	80
DELETESWEEPS	80
Detrending (Linear Detrending)	80
DETREND	81
DETREND_EX	81
EXCLUDEFORDETREND	81
RESETFORDETREND	81
ENABLEOVERWRITEPROMPT	82
EPOCH	82
EPOCH_EX	83
ERBP	83

ERCOH	84
Exporting/Importing ASCII/EDF Files	85
EXPORTAVG	85
EXPORTAVG_EX	86
EXPORTAVG_EX2	87
EXPORTCNT	87
EXPORTCNT_EX	87
EXPORTCNT_EX2	88
EXPORTCOH_EX	89
EXPORTCOH_EX2	89
EXPORTEDF	89
EXPORTEDF_EX	90
EXPORTEEG	90
EXPORTEEG_EX	90
EXPORTEEG_EX2	91
IMPORTAVG	92
IMPORTCNT	92
IMPORTEEG	92
IMPORTEEG_EX	93
EXTRACT	93
EXTRACTBLOCK	94
Filtering data files	94
FILTER	94
FILTER_EX	95
EXCLUDEFORFILTER	95
RESETFORFILTER	95
FSP	96
FSP_EX	96
Get/Set commands	97
GETCOMP/DELETEALLCOMP	97
GETCURRENTEPOCHINDEX/SETCURRENTEPOCHINDEX	97
GETEPOCHCOUNT	98
GETEPOCHINFO/SETEPOCHINFO	98
GETEVENTCOUNT	100
GETEVENTINFO/SETEVENTINFO	100
GETMARKERCOUNT	102
GETMARKERINFO	103
GETPOINTDATA	104
GFP	105
IMPORTEVENT	105
INSERTRESPONSEEVENT	105
INSERTSTIMEVENT	106
INTEGRATEWAVEFORM	108
LDR	108
LOAD3DD	109
MERGEVET	109
Open/Save Commands	109
OPENFILE	109
SAVEAS	110
SAVEAS_EX	110
Peak Detection	110
EXCLUDEFORPEAKDETECTION	110
PEAKDETECTION	111
PEAKDETECTION_EX	111

RESETFORPEAKDETECTION	112
SAVEPEAK	112
PRINT	113
QRS	113
RECTIFY	114
RECTIFY_EX	114
REFER	114
REJECTBLOCK	115
REMOVECOMPAREELECTRODES	115
REMOVEEVENT	116
RESPWIN	116
SAVEEVENT	117
SAVEEVENT_EX	117
SELECTFILE	118
Smoothing Data	118
EXCLUDEFORSMOOTH	118
RESETFORSMOOTH	119
SMOOTH	119
SMOOTH_EX	119
SMOOTH_EX2	120
SNR	120
SORT	120
SPATIALFILTER	121
SPATIALSVD	122
Spectral analyses	122
FORWARDFFT/INVERSEFFT	122
SPECTRUM	123
SPLINEFIT	124
Statistical Comparisons	124
CROSSCORR	124
CROSSCORR_EX	125
INTRACLASSCORR	125
INTRACLASSCORR_EX	125
PAIRED	126
PEARSONS	126
PEARSONS_EX	127
TSCORE	127
ZSCORE	127
SUBTRACT	128
SUBTRACTAVG	128
TRUNCATECNT	129
UPDATECHANGES	129
VOLTAGETHRESHOLD	129
Windowing Data Files	130
EXCLUDEFORWINDOWDATA	130
RESETFORWINDOWDATA	130
WINDOWDATA	130
WINDOWDATA_EX	130
WRITENUMSWEEPS	131

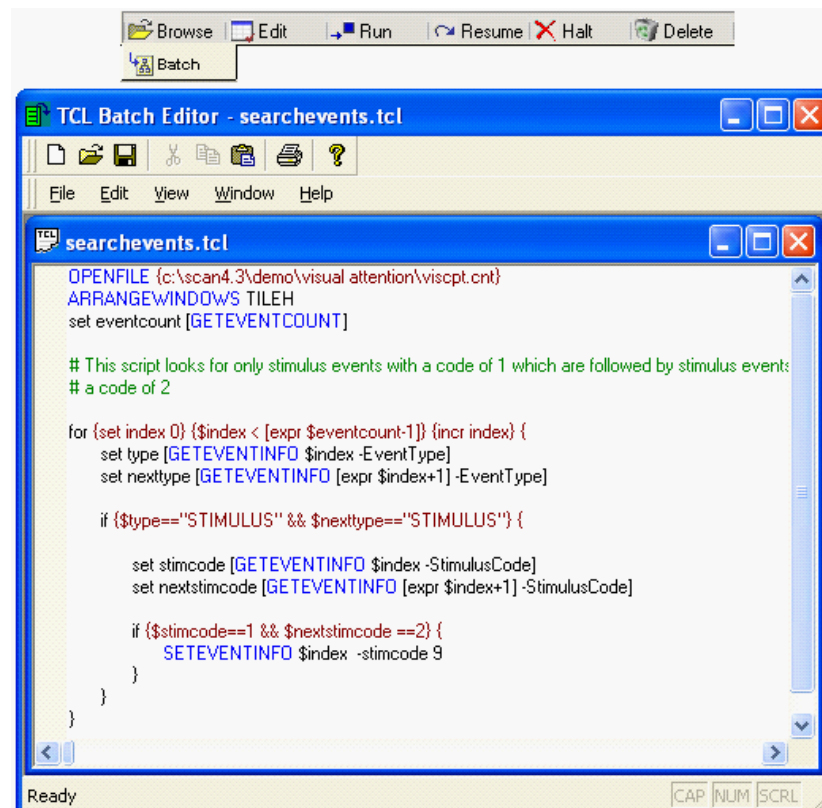
Part IV Advanced TCL Scripts 132

Part V Quick Reference Guide 137

Index**0**

1 Tcl Batch

Tcl Batch Commands User Manual



Compumedics USA, Inc.

6605 West W.T. Harris Blvd., Suite F
Charlotte, NC 28269
USA
Telephone: 877-717-3975 (8am – 5pm EST)



Compumedics Germany GmbH
Heussweg 25
20255 Hamburg, Germany
Telephone: +49 40 40 18 99 41
Fax: +49 40 40 18 99 49

Internet: sales@neuroscan.com
techsup@neuroscan.com
www.neuroscan.com



1.1 Contact Information

For Technical Support with BATCH files, we strongly recommend you send your questions via e-mail to techsup@neuroscan.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.neuroscan.com>).

Copyright © 2009 - 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 Compumedics/Neuroscan.

7318A Tcl Batch45

2 Introduction

Tcl BATCH Files

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 created in a text editor (the Batch 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).


2.1 Executing BATCH Files in SCAN

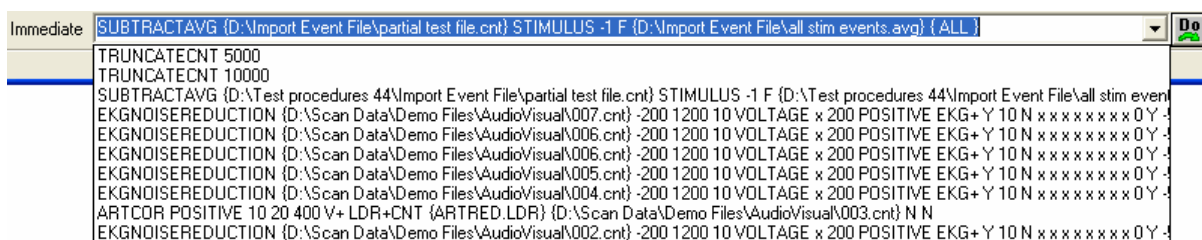
We will assume you have SCAN 4.3 (or newer version) installed and working, and that you have a basic understanding of the operations of EDIT and ACQUIRE. You can create BATCH files for use in either program: offline in EDIT, or online in ACQUIRE. Generally, most BATCH processing is performed offline, so we will start there. Go into the EDIT program, and see the **Batch** and **Immediate** Toolbars. If you do not see them, go to **View** →

Toolbars and enable them. Click the **History** icon  on the Toolbar, if needed. These all have uses in creating or running BATCH files.

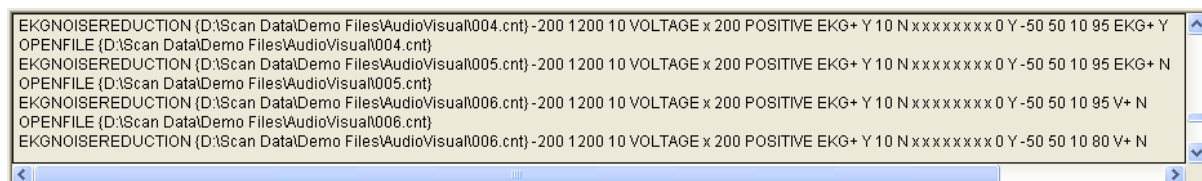
Batch Toolbar. The BATCH controls are used to retrieve, create, modify, and execute batch files. Use the Browse button  as needed to select the Tcl file you wish to run.

The  button displays the Batch Editor, a text editor designed especially for Tcl BATCH files. The  button executes the highlighted BATCH file. If you are using the PAUSE or REVIEW commands, the program will stop at that point until you click the  button. The  button will terminate the BATCH file.

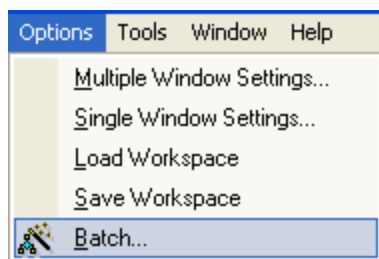
Immediate tab. The Immediate field allows you to execute a single line BATCH command. The most recent operations are displayed in the pull-down list field. Select a single command and click  to execute it immediately (you must have the data file displayed). You can also copy commands from the Immediate lines and paste them into the Batch Editor when creating BATCH files.



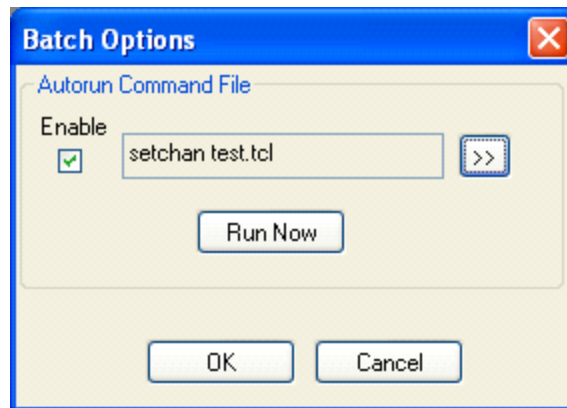
History field. The History field contains an accumulating list of operations you have performed. Some commands will be listed here that are not included in the Immediate field (such as, OPENFILE commands). These can be copied (**highlight** → **right click** → **Copy**) and pasted into the Batch Editor when creating BATCH files.



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



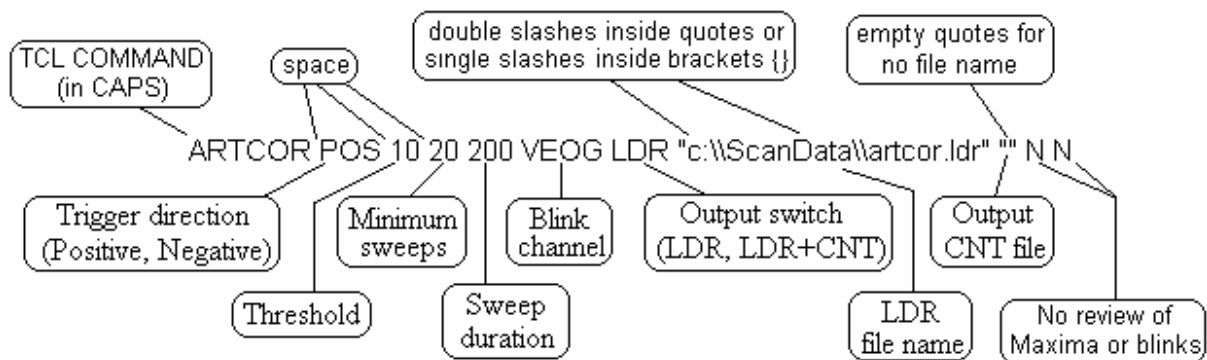
Selecting this option displays the Autorun feature controls.



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.

2.2 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:\Scan Data\p300.eeg"
FILTER BANDPASS ZERO 1.5 24 50 24 0 0 0 0 "c:\Scan
Data\filtp300.eeg"
OPENFILE "c:\Scan Data\filtp300.eeg"
DETREND ENT 0 0 "c:\Scan Data\detrendp300.eeg"
```

```
OPENFILE "c:\\Scan Data\\detrendp300.eeg"  
AVERAGE T N N "" 0 0 0 "" "c:\\Scan Data\\averagep300.avg"  
CLOSEALL
```

2.3 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:\\Scan Data`, 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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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:\Scan Data\secondfile.avg} {P3 P4 C3 C4} {c:\Scan
Data\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:\Scan Data\file1.avg} {c:\Scan Data\file2.avg} {c:\Scan
Data\file3.avg} }
```

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

```
DELETE {"c:\\Scan Data\\file1.avg" "c:\\Scan Data\\file2.avg" "c:\\Scan
Data\\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

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).

2.4 Creating BATCH Files

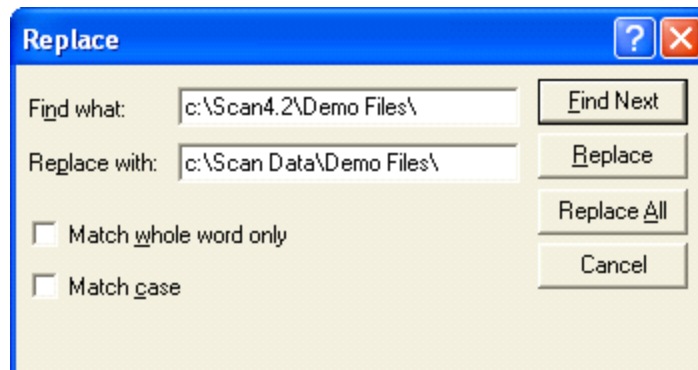
The Tcl Batch Editor  provides the easiest means for creating or modifying BATCH files, and it is strongly recommended over other text editors (such as Word, Wordpad and Notepad). The Tcl BATCH Editor was designed especially for Tcl BATCH files. Click the  icon to open the program. It is very easy to use, and most of its functionality is obvious and self-explanatory. One thing you will notice is that the text appears in colors.

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

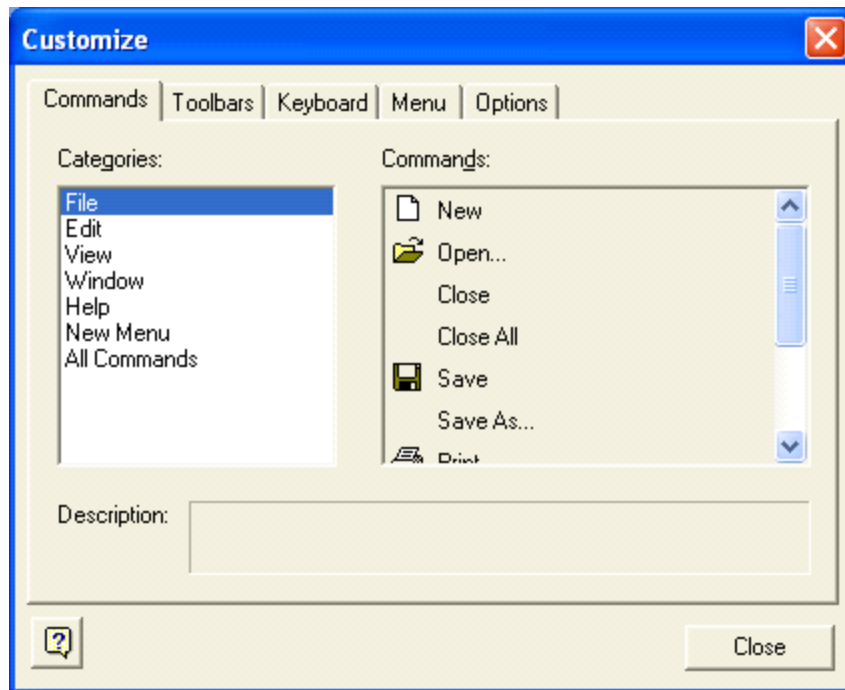
OPENFILE "$path\\cor_1.avg"
INSTRUCT "Follow the P&C test procedures for Time Domain Test-1."
PEARSONS_EX {c:\\scan43 test files\\Pearsons\\cor_2.avg} -25 99.5 {c:\\sr
CLOSEALL
```

Comments are in green, Tcl commands are in blue, text strings are in red, and so forth. This is purely for your 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 you have a file open (New or existing), the **Edit** option appears on the Main Menu bar. 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.

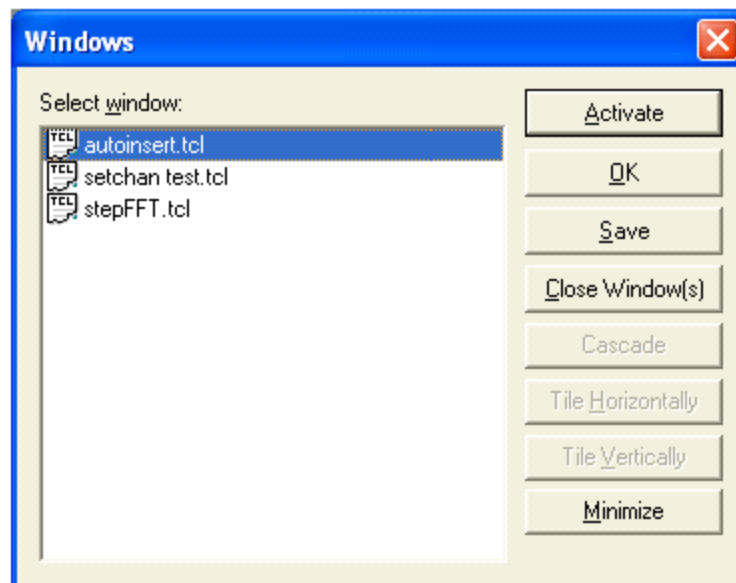


You have flexibility in the appearance of the Toolbar in the Editor. Click **View** → **Toolbar**, and then **Customize** to see the following display. The tabs contain various options for modifying the Toolbars, creating Shortcut keys, and various additional options.




Another useful option in the Batch Editor is found by clicking **Window** → **Windows**. This 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 a convenience for managing multiple open files.



In practice, the BATCH files are created in the editor, and it is 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 selected the file in EDIT, you do 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** 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.

2.5 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:\\Scan Data\\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 valuetoad 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:\\Scan Data\\"
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` 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):

break Terminates the innermost nested looping command.

continue Terminates the current iteration of the innermost looping command and goes on to the next iteration of that command.

for init test reinit body Executes `init` as a Tcl script. Then evaluates `test` 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`.

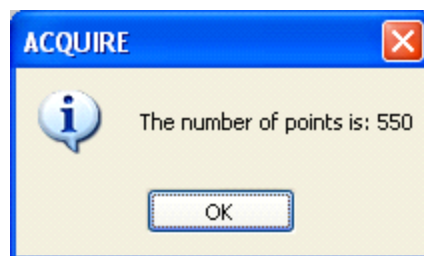
The following are commonly used arithmetic operators (abstracted from Tcl and the Tk Toolkit, JK Ousterhout, 1994; in the Results column, 1 means true and 0 means false).

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

2.6 Get and Set Commands

In the list of commands presented below, 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:\\Scan Data\\vcp.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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vcp.avg"
```

```
set start [LATENCYTOPOINT 97]
set stop [LATENCYTOPOINT 104]
set data [GETPOINTDATA $start $stop {OZ}]
INSTRUCT $data
CLOSEALL
```

2.7 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:\\Scan Data\\Batch Examples\\$a"
```

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

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

You can use multiple variable substitutions in the same command.

```
set path "c:\\Scan Data\\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:\\Scan Data\\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:\\Scan Data\\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`, which 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:\\Scan Data\\Batch Examples\\file1.avg*). Now, the `OPENFILE` command is used to open the `$username` file, and the `BASECOR` operation is performed. In this 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**

BATCH commands 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:\Scan Data\data\file1.eeg} {c:\Scan Data\data\file2.eeg} {c:\Scan
Data\data\fileN.eeg}} {c:\Scan Data\data\outputfile.eeg}
```

Braces MUST be used with this command (not quotes). That will run without any problems; however, consider the same case where variable substitutions are present:

```
set path {c:\Scan Data\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:\\Scan Data\\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 result will be a file named `c:\Scan Data\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:\\Scan Data\\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.

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:\\Scan Data\\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.

Multiple Data Files: 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 and process several data files. For example, the `glob` command can be used with the `foreach` command to retrieve all CNT files in a specified folder and then perform the same set of operations on them. It is the most flexible of the ways to apply the batch file to multiple data files, since there are no restraints on the file names (it uses only the file extension).

The substituted variables are defined first in the batch file below. These include all elements of the output and input data files. For example, \$path\$fileroot\$tag1\$cnt replaces D:\\Scan Data\\Demo Files\\CNT Files\\<root file name>_filt.cnt. The glob command is basically saying to take each of the CNT files in the designated folder and put them in a "list". foreach takes each of those files in the list and calls it "file". PICKOFFNAMEONLY extracts the root file name of the file (all but the .extension).

The DELETE commands at the end delete the filtered CNT file that is created, as well as the temporary CNT file that is created when using the FIR filter with continuous data files. These commands are added simply to remove the additional CNT files in case you need to rerun the batch file (as during debugging). Otherwise, since they are CNT files, they will be included in the glob list and the batch file will process them as well the next time you run it.

```
set path "D:\\Scan Data\\Demo Files\\CNT Files\\"
set tag1 "_filt"
set tag2 "_base"
set cnt ".cnt"
set eeg ".eeg"
set avg ".avg"

set list [glob -directory $path *.cnt]
foreach file $list {

    set fileroot[PICKOFFNAMEONLY $file]
    OPENFILE $file
    FILTER_EX LOWPASS ZEROPHASESHIFT x x 30 48 x x x N FIR { ALL }
    $path$fileroot$tag1$cnt
    OPENFILE $path$fileroot$tag1$cnt

    CREATESORT SORT1
    SORT1 -TypeEnabled T
    SORT1 -TypeCriteria 1
    EPOCH_EX PORT_INTERNAL "" N -100 500 N Y Y N N SORT1
    $path$fileroot$eeg
    DELETESORT SORT1
    OPENFILE $path$fileroot$eeg

    BASECOR_EX2 PRESTIMINTERVAL x x N N { ALL } $path$fileroot$tag2$eeg
    OPENFILE $path$fileroot$tag2$eeg

    ARTREJ_EX REJCRITERIA Y x x Y -50 50 Y Y { Fp1 Fp2 FC3 }

    AVERAGE_EX TIME Y AMPLITUDE 10 COSINE PRESTIMINTERVALNOISE x
    x POSTSTIMINTERVALSIGNAL x x NULL $path$fileroot$avg
    OPENFILE $path$fileroot$avg
```

```
SCALE -15 15
MAXIMIZE
ZOOMINFZ
PAUSE 3000
CLOSEALL
DELETE [list $path$fileroot$tag1$cnt]
DELETE {{D:\Scan Data\Demo Files\CNT Files\TempScan42Filter.cnt}}
}
```

All CNT files in the designated folder will be processed in the loop, one file at a time.

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.

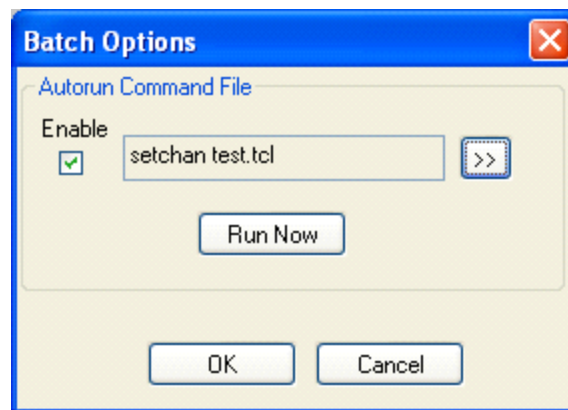
`set varName ?value?` - If `value` is specified, sets the value of variable `varName` to `value`. In any case the command returns the (new) value of the variable.

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

2.8 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:\\Scan Data\\Batch Examples\\" $file]
}
```

Then the following command:

```
OPEN viscpt.cnt
```

can be used as a shortcut to replace the longer

```
OPENFILE "c:\\Scan Data\\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:

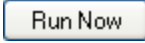

```
proc C {} {
    CLOSEALL
}
```

Now, simply typing C (note: these are case sensitive) in the Immediate window and pressing ENTER 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.

2.9 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 **ARRANGEWINDOVS** 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 `\Scan Data\Batch Examples` folder. The "*Example 8-1.tcl*" file demonstrates the uses of several of the "GET" commands.

```

set path "c:\\Scan Data\\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 EPOCH 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:\\Scan Data\\Demo Files\\VisualAttention\\viscpt.cnt}
ARTCOR pos 10 20 400 "VEOG" LDR+ {c:\\Scan Data\\Demo
Files\\VisualAttention\\viscpt.ldr}{c:\\Scan Data\\Demo
Files\\VisualAttention\\viscptCORR.cnt} Y Y
OPENFILE {c:\\Scan Data\\Demo Files\\VisualAttention\\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:\Scan Data\Demo
Files\VisualAttention\viscpt.eeg}
OPENFILE {c:\Scan Data\Demo Files\VisualAttention\viscpt.eeg}
AVERAGE TIME n n "" 0 0 0 RareSort {c:\Scan Data\Demo
Files\VisualAttention\RAREtype1.avg}
OPENFILE {c:\Scan Data\Demo Files\VisualAttention\RAREtype1.avg}

# The "focus" is now returned to the viscpt.eeg file
SELECTFILE {viscpt.eeg}
AVERAGE TIME n n "" 0 0 0 FreqSort {c:\Scan Data\Demo
Files\VisualAttention\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:\Scan Data\Demo Files\VisualAttention\FREQtype2.avg} {c:\Scan
Data\Demo Files\VisualAttention\difference.avg}
OPENFILE {c:\Scan Data\Demo Files\VisualAttention\difference.avg}
ZOOMIN PZ
PRINT
CLOSEALL

```

**Note**

The *OPENFILE* command is used 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**

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 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).

2.10 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:\Scan Data\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:\Scan Data\Batch Examples\sample.tcl.

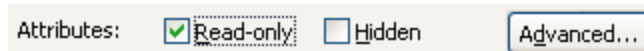
3 Tcl BATCH Commands

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.

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.

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 or COH 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 **Miscellaneous** panel in ACQUIRE).



"_EX" or EXtended Commands

Beginning with SCAN 4.3, there are several "_EX", or EXtended commands, including for example, BASECOR_EX, EXPORTAVG_EX2, and various 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. For instance, some of the new commands 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. When the "_EX" commands themselves are subsequently modified, the new command will have "_EX2", as in EXPORTEEG_EX2. The EDIT program will **"Auto-write"** the most recent command in all cases.

In the descriptions of the commands below, the earlier versions of the commands will be in *gray*. The commands will still function, but the latest version of the command will be the one created in the History and Immediate fields.

3.1 Tcl Commands for ACQUIRE and EDIT

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

3.1.1 ADDDISPLAYFILTER / REMOVEDISPLAYFILTER

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	Band Stop start frequency
7	float	Band Stop stop frequency
8	int	Notch poles
9	Boolean	Rectify
10	list	Channels to be filtered (or "All")


Example. ADDDISPLAYFILTER BANDPASS 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.

3.1.2 ADDWAVEBOARD

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
---	------	--

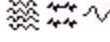
Example. ADDWAVEBOARD {FZ CZ PZ OZ}. The indicated channels will be sent to the waveboard. A more complete example is as follows, where channels from two equivalent data files are sent to the Waveboard:

```

OPENFILE "c:\\Scan Data\\Batch Examples\\sepblk.avg"
ADDWAVEBOARD {C3 C4}
OPENFILE "c:\\Scan Data\\Batch Examples\\sepnoblk.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

```

3.1.3 ARRANGEWINDOWS

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:\\Scan Data\\Batch Examples"
for {set n 0} {$n < 3} {incr n} {
    OPENFILE "$path\\sepblk.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
    }
}

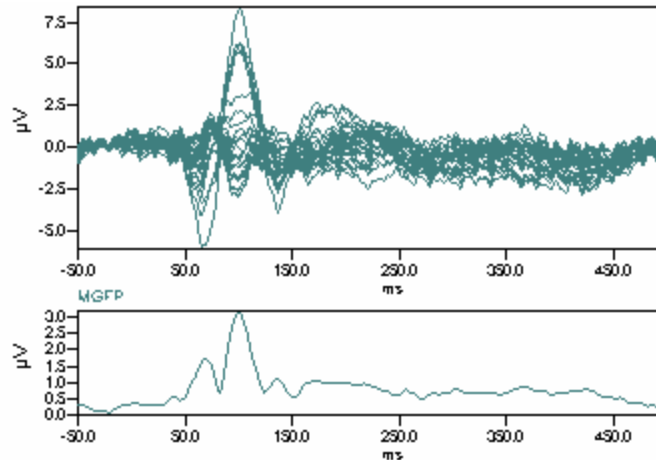
```

3.1.4 BUTTERFLYPLOT


BUTTERFLYPLOT - 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. BUTTERFLYPLOT Y. The data will be displayed in the butterfly plot mode.



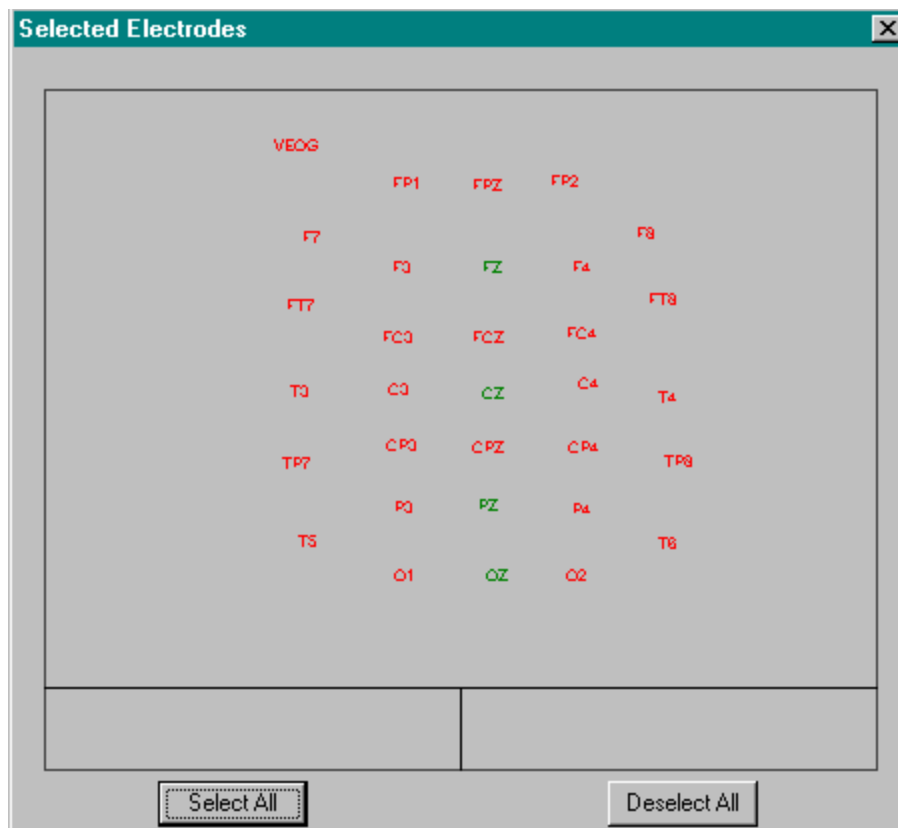
3.1.5 CHANLIST

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

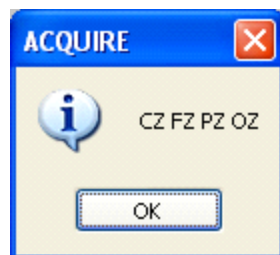
Example. CHANLIST {FZ CZ PZ OZ} "Selected Electrodes". You will see a display such as the following.



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

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


The contents of the list will be displayed:



3.1.6 Data Scaling Commands

These commands are used to scale the data displays automatically or manually.


3.1.6.1 AUTOSCALE

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.

3.1.6.2 SCALE


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	double	Minimum voltage
2	double	Maximum voltage

Example. `OPENFILE "c:\\Scan Data\\Batch Examples\\MYDATA.avg"`
 `SCALE -10 10`
 `SAVEAS "c:\\Scan Data\\Batch Examples\\MYDATA.avg"`

In this example, the file MYDATA.avg is recalled, scaled to $\pm 10 \mu\text{Vs}$, and overwritten to disk.

3.1.6.3 SCALEX

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	double	Minimum time point
2	double	Maximum time point


Example. `OPENFILE "c:\\Scan Data\\Batch Examples\\MYDATA.avg"`
 `SCALEX 0 400`
 `SAVEAS "c:\\Scan Data\\Batch Examples\\NEWDATA.avg"`

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

3.1.7 File Management Utility Commands

These are some basic commands that are used throughout the online and offline programs.

3.1.7.1 DELETE


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:\Scan Data\Batch Examples\badfile1.cnt} {c:\Scan Data\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]
```

3.1.7.2 NEWNAME


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:\\Scan Data\\oldname.eeg" "c:\\Scan Data\\newname.eeg".
```

3.1.7.3 RENAME

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".

3.1.8 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.

3.1.8.1 GETBASELINECOR/SETBASELINECOR

GETBASELINECOR / SETBASELINECOR - Get/Set Baseline Display

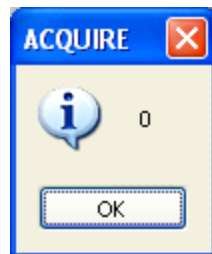
Correction.  Use this command to query the baseline correction (display only) status of the current working file.

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

The returned information will appear as shown.

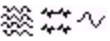


SETBASELINECOR

1 Boolean Enable

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

3.1.8.2 GETCHANATTRIBUTE

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

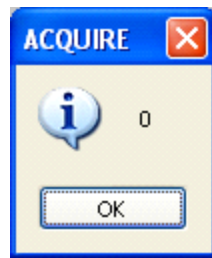
Return. The command will return a 1 if the attribute is set, or a 0 if it is not.

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

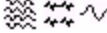
Example. GETCHANATTRIBUTE "CZ" -Bad. The program will return a 1 or a 0, depending on whether CZ is set as a Bad channel or not. The following script shows how to view the returned value.

```
OPENFILE "c:\\Scan Data\\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.



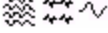
3.1.8.3 SETCHANATTRIBUTE

SECHANATTRIBUTE - 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.

3.1.8.4 GETCHANLABEL

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

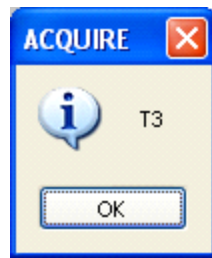
Return. The command will return the channel label.

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

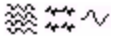
Example. GETCHANLABEL "12". The command will return the label for channel 12. Note that this is a zero-based index, meaning that the first electrode is 0, not 1. The T3 label that is returned is the 13th electrode in the setup file. To see the returned string, use a script similar to the following:

```
set path "c:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vep.avg"
set label [GETCHANLABEL "12"]
INSTRUCT $label
```

You will see the label for channel 13.



3.1.8.5 GETCHANNELINDEX

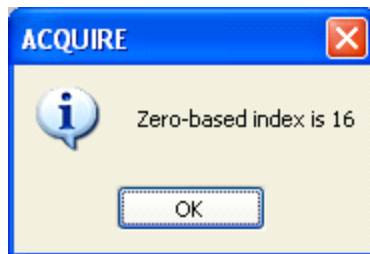
GETCHANNELINDEX - Get Channel Index.  This command will return a zero-based index for the selected channel. See also **GETCHANLABEL**.

1 string Electrode label.

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

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

The returned value is displayed:



3.1.8.6 GETDATE/GETTIME

GETDATE / GETTIME - Get the current date and time. Use these commands to return the current date and time (using the computer clock).

Return. The commands will return the current date and time.

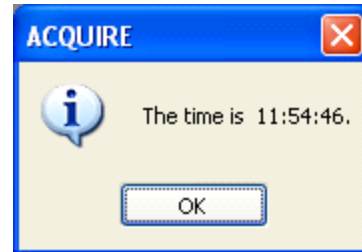
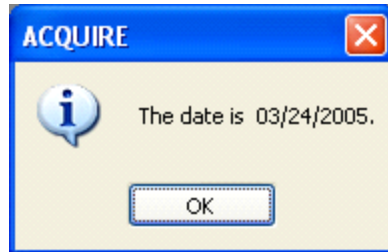
1 Boolean Use formatting

Example. GETDATE Y. The current date will be returned in the form mm/dd/yyyy (and hour:min:sec). If you use the negative Boolean setting, the information will have no formatting (mmddyyyy and hourminsec).

```
set date [GETDATE Y]
INSTRUCT "The date is $date."
set time [GETTIME Y]
```

INSTRUCT "The time is \$time."

The returned values are as shown.



3.1.8.7 GETDISPLAYINVERTED/SETDISPLAYINVERTED

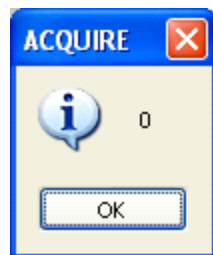
GETDISPLAYINVERTED / SETDISPLAYINVERTED - Get/Set Display Inversion.  These commands are used to determine the current status of the Invert Polarity command (GETDISPLAYINVERTED), or to set the display polarity (SETDISPLAYINVERTED).

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:\\Scan Data\\Batch Examples\\vcp.avg"
set invert [GETDISPLAYINVERTED]
INSTRUCT $invert
```

The returned value is displayed as shown.



SETDISPLAYINVERTED

1 Boolean Invert

Example. SETDISPLAYINVERTED Y. The polarity will be inverted.

3.1.8.8 GETDISPLAYPAGE / SETDISPLAYPAGE

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.

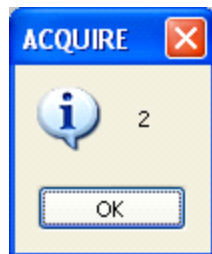
Example. GETDISPLAYPAGE. This command has no parameters. A simple example for seeing the returned value is as follows (after retrieving a file and displaying a display page):

```
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).



3.1.8.9 GETINPUT

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.

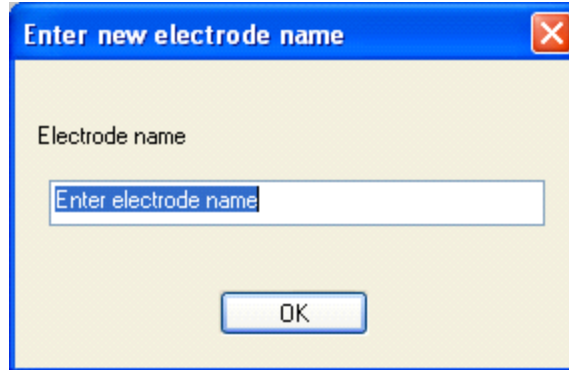
1	string	Title
2(optional)	string	Prompt
3(optional)	string	Edit default

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:\\Scan Data\\Batch Examples\\vcp.avg"
set newlabel [GETINPUT "Enter new electrode name" "Electrode name"
"Enter electrode name"]
```

```
# Enter Czcd for the new label  
RENAME "CZ" "$newlabel"  
SAVEAS "c:\\Scan Data\\Batch Examples\\newvcp.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 *vcp.avg* file), but it will not be applied unless you save that file with the modification.

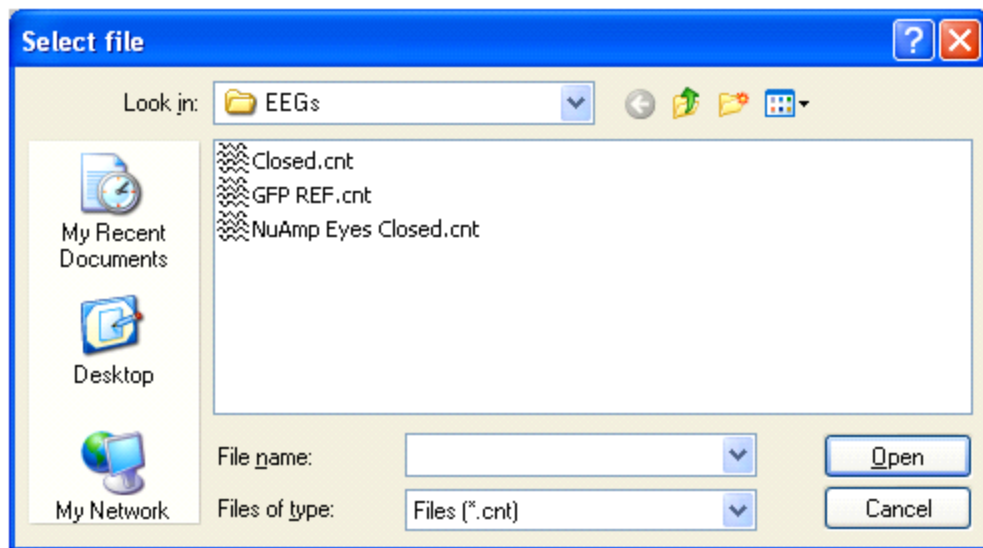


3.1.8.10 GETINPUTFILE


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:\\Scan Data\\Demo Files\\EEGs\\". 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 precede with the rest of the BATCH commands.



3.1.8.11 GETNUMCHANS

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.

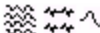
Example. GETNUMCHANS appears on a line by itself, and uses no parameters. The returned value is the number of channels. To see the returned value, use a script similar to the following:

```
set path "c:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set number [GETNUMCHANS]
INSTRUCT $number
```

The number of channels will be displayed.



3.1.8.12 GETNUMDISPLAYPAGES

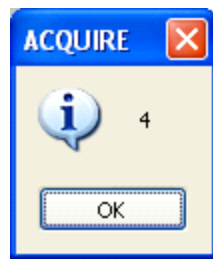
GETNUMDISPLAYPAGES - Get Number of Display Pages.  This command will return the number of Display Pages in any type of data file.

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

The number of display pages will be displayed.



3.1.8.13 GETNUMPOINTS

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

The number of data points will be displayed.



3.1.8.14 GETNUMSWEEPS

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.

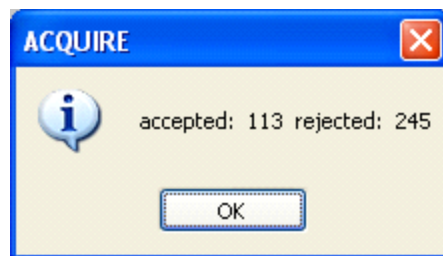
1 defined value Parameter (-Accepted, -Rejected)

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:\\Scan Data\\Batch Examples\\p300.eeg"
set acc [GETNUMSWEEPS -Accept]
set rej [GETNUMSWEEPS -Reject]

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

The numbers of Accepted and Rejected sweeps are returned.

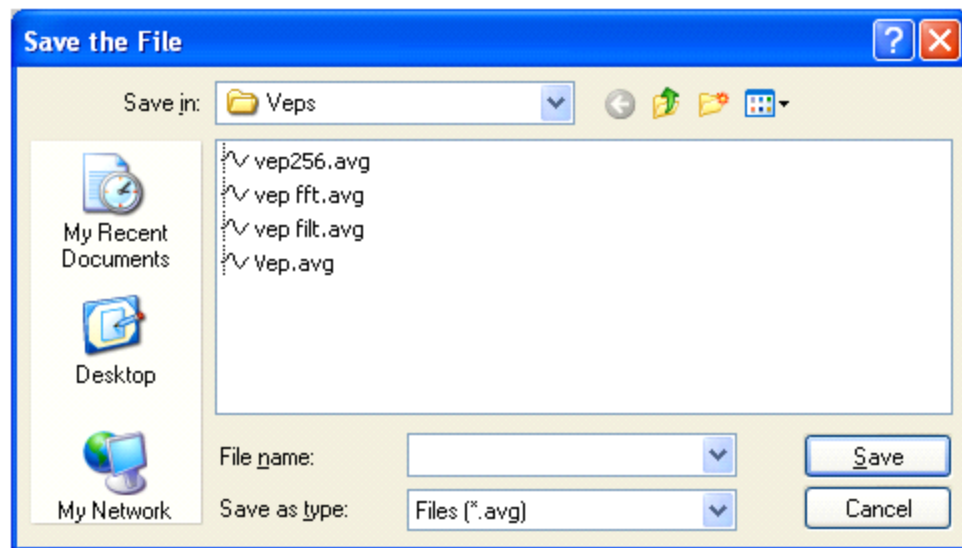


3.1.8.15 GETOUTPUTFILE

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:\\Scan Data\\Demo Files\\Veps". 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.



3.1.8.16 GETSWDISPLAYSPEED / SETSWDISPLAYSPEED

GETSWDISPLAYSPEED / SETSWDISPLAYSPEED - Get/Set Display

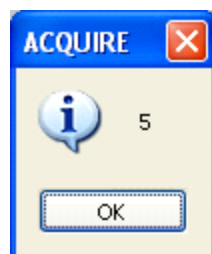
Speed. ⓘ These commands are used to return the current number of seconds in the display (the display speed), or set the number of seconds in the display (CNT files only).

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:\\Scan Data\\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.


3.1.8.17 GETSWEEPMAX

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.

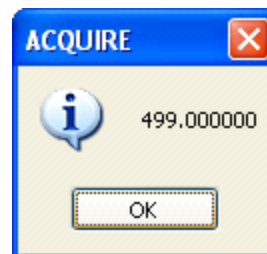
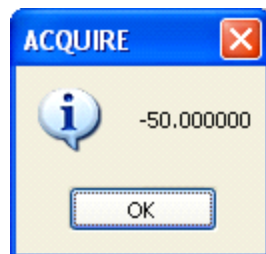
3.1.8.18 GETSWEEPMIN

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



3.1.9 HERTZTOPOINT

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.

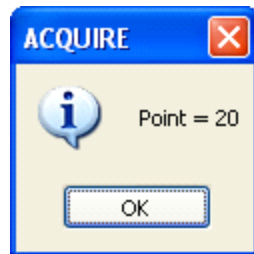
Return. The zero-based, frequency bin number nearest the selected Hz value is returned.

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:\\Scan Data\\demo\\EEGs\\closedfft.avg"
set point[HERTZTOPOINT 10]
INSTRUCT "Point = $point"
```

In this example, "20" is returned.



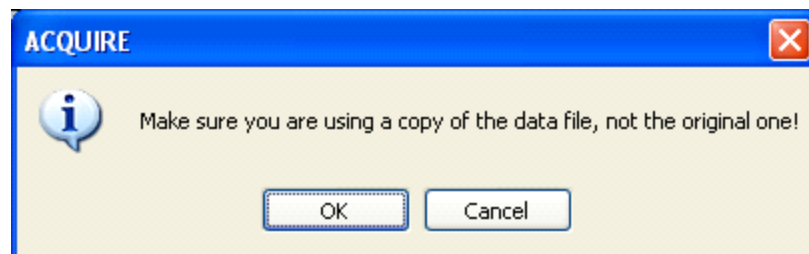
3.1.10 INSTRUCT

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).

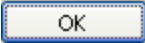
1 string Instruction
2(optional default =OK) defined value Type (OK, OKCANCEL, RETRYCANCEL, YESNO, YESNOCANCEL)

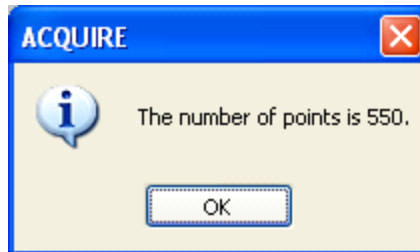
Example. **INSTRUCT** "Make sure you are using a copy of the data file, not the original one!" **OKCANCEL**. The message will be displayed, with the OK and Cancel options available to the user.



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

```
OPENFILE "c:\\Scan Data\\Batch Examples\\vep.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).



3.1.11 ISOFFLINE / ISONLINE

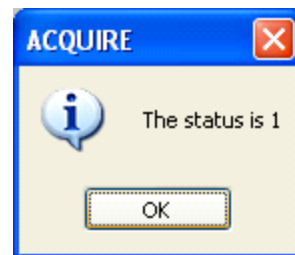
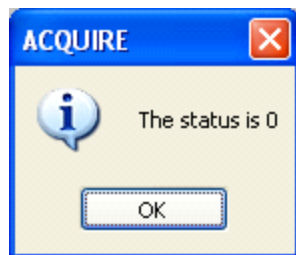
ISOFFLINE / ISONLINE - Online or Offline. These commands are used primarily in conjunction with the **autorun** feature. You may want to have a different set of functions loaded depending on whether you are online (in ACQUIRE) or offline (in EDIT). The commands will return either a 0 or 1, depending 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.



3.1.12 LATENCYTOPOINT

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

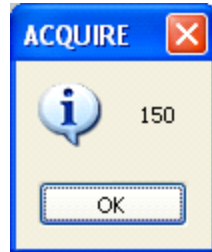
calculate the index numbers, or, you can use **LATENCYTOPOINT**. Enter the latency of a data point, and the command will return the index number of the point. See also **POINTTOLATENCY**.

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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set point [LATENCYTOPOINT 100]
INSTRUCT $point
```




3.1.13 MAXIMIZE

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.

3.1.14 MINIMIZE

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.

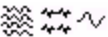
3.1.15 PAUSE

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.

3.1.16 "PICKOFF" commands

"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:\Scan Data\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
---	--------	--------------------

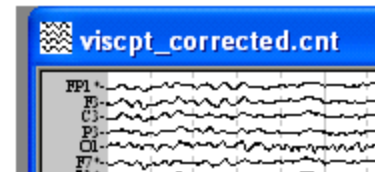
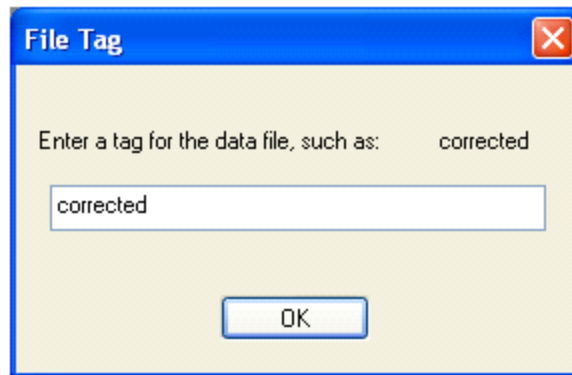
Use is demonstrated in the following script, where the name and path for the output file are created by "picking off" information from the "path" variable:

```
set path "c:\\Scan Data\\Demo Files\\VisualAttention\\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
```

The variable "OutFile" consists of the path, the original file name, an underscore, the added part of the file name (the "tag"), and the file extension.



3.1.17 POINTTOHERTZ

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 is the frequency bin number. The command is used with frequency domain AVG and EEG files only. See also **HERTZTOPOINT**.

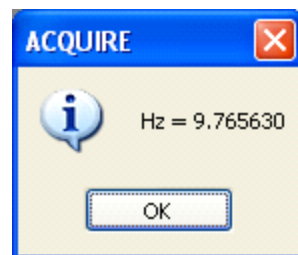
Return. The Hz value nearest the selected zero-based, frequency bin number is returned.

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.



3.1.18 POINTLATENCY

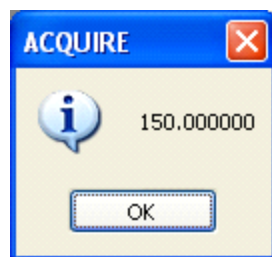
POINTLATENCY - Point Index to Latency.  This command is the complement to the **LATENCYTOPOINT** command, and is used to return the latency nearest to the data point you specify (EEG and AVG files only). Enter the zero-based index of the data point, and the command will return the latency (in milliseconds) 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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vep.avg"
set latency [POINTTOLATENCY 200]
INSTRUCT $latency
```




3.1.19 PRINTWAVEBOARD

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.

3.1.20 READPOS/SAVEPOS

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:\\Scan Data\\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.

The following batch file demonstrates the SAVEPOS and READPOS commands.

```
OPENFILE {C:\\Scan Data\\Demo Files\\Veps\\Vep.avg}
SAVEPOS {C:\\Scan Data\\Demo Files\\Veps\\positions.asc}
```

```

INSTRUCT "Reposition the electrode windows manually."
PAUSE
INSTRUCT "Continue the batch file to return the positions to the original
placements."
PAUSE
READPOS {C:\Scan Data\Demo Files\Veps\positions.asc}
PAUSE
CLOSEALL

```

3.1.21 REM

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.

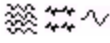
```

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.

3.1.22 RESTORE


RESTORE - Restore Window.  This command performs the standard Windows Restore function (to Restore a Minimized display). Applying it will restore the display of the window that has the focus. Use SELECTFILE to change the focus to the desired window, as needed.

Example. RESTORE. No additional parameters are used.

3.1.23 REVIEW

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:\Scan Data\Demo Files\P300s\p300.eeg}
set epochcount [GETEPOCHCOUNT]
for {set index 0} {$index < $epochcount} {incr index} {
  SETCURRENTEPOCHINDEX $index
  REVIEW 1000
}

```

3.1.24 SETSPECTDISPLAYSTYLE

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.

3.1.25 Subject Information Commands

These commands are relevant for accessing, modifying and saving subject information.


3.1.25.1 READSUB

READSUB - Read Subject Information File. The READSUB command will load a subject information file. It has the same function as the  button on the Subject dialog screen. 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:\\Scan Data\\Batch Examples\\subjectfile.sub".

3.1.25.2 SAVESUB

SAVESUB - Save Subject Information File. The SAVESUB command will save a subject information screen to disk. It has the same function as the  button on the Subject dialog screen. 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:\\Scan Data\\Batch Examples\\subjectfile.sub".

3.1.25.3 SUBJECT

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.

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

3.1.26.1 INSERT

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:\\Scan Data\\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).

3.1.27 WRITELOG

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:\\Scan Data\\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:\\Scan Data\\Demo Files"
OPENFILE "$path\\sample file.eeg"
AVERAGE T N N "" 0 0 0 "" "$path\\average file.avg"
WRITELOG "$path\\mylog.log" "Text entered here will appear in the LOG file."
WRITELOG "$path\\mylog.log" "Text written to the same file will be added at the
end of the file."
CLOSEFILE
```

The log file (as seen in a text editor, such as Wordpad) is seen as:

```
Text entered here will appear in the LOG file.
Text written to the same file will be added at the end of the file.
```

3.1.28 ZOOM Commands

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

3.1.28.1 TOGGLEZOOM

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. Repeat to minimize the display.

3.1.28.2 ZOOMIN

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.

3.1.28.3 ZOOMOUT

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.

3.2 Tcl Commands for ACQUIRE


The following batch commands are used in the ACQUIRE program only.

3.2.1 CALIB

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.

3.2.2 CLEARAVG

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.

3.2.3 DODCCORRECT

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.

3.2.4 DOIMPEDANCE

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.

3.2.5 GETAST

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 .

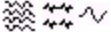
3.2.6 HIGHIMPEDANCE

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).

3.2.7 PAUSEDISPLAY

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 the to next command: **PAUSEDISPLAYY**, 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
```

3.2.8 RESTART

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.

3.2.9 STARTACQUISITION

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.

3.2.10 STARTRECORDING

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).

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

Example. STARTRECORDING "c:\\Scan Data\\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.

3.2.11 STOPACQUISITION

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.

3.2.12 STOPRECORDING

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.


3.2.13 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:\\Scan Data\\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
```

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.

3.2.14 VIEW

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).




3.2.15 Discontinued commands for ACQUIRE

Beginning with SCAN 4.4, the batch commands for configuring and enabling online Blink Reduction and online EKG Noise Reduction have been discontinued. These operations generally require real-time interaction to set the parameters, therefore it does not make sense to have batch commands for them. If you have batch files to control your acquisition, and you are using these artifact reduction procedures, then we recommend that you place PAUSE commands at the appropriate places in the batch files, and then set/apply the corrections as needed. (You need a Toolbox license for the online blink correction, and a MagLink RT license for the online EKG/BCG correction. There are no batch commands for the online fMRI artifact correction).

3.3 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.

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 or COH files.

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 and Immediate fields. From there you can copy the line into the Batch Editor.

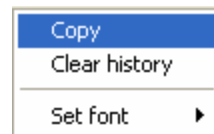
For example, open a CNT file and Epoch it in P&C mode. In this example, we sorted for type codes of 1. The required lines are created automatically, including the "sort" file lines (your SORT files will have a different number), and displayed in the **History** field.

```

OPENFILE {C:\Scan Data\Demo Files\VisualAttention\viscpt.cnt}
CREATESORT SORT72
SORT72 -TypeEnabled T
SORT72 -TypeCriteria 1
EPOCH_EXPORT_INTERNAL "" N -100 996 N Y Y N N SORT72 {C:\Scan Data\Demo Files\VisualAttention\viscpt.eeg}
DELETESORT SORT72
OPENFILE {C:\Scan Data\Demo Files\VisualAttention\viscpt.eeg}

```

These lines can then be copied and pasted into the BATCH file you are creating by highlighting the lines, then clicking the *right mouse* in the History region and selecting the **Copy** option. (You can also Clear the field of all listed commands, as well as change the size of the Font by selecting Set font and then selecting either Big, Medium, or Small.



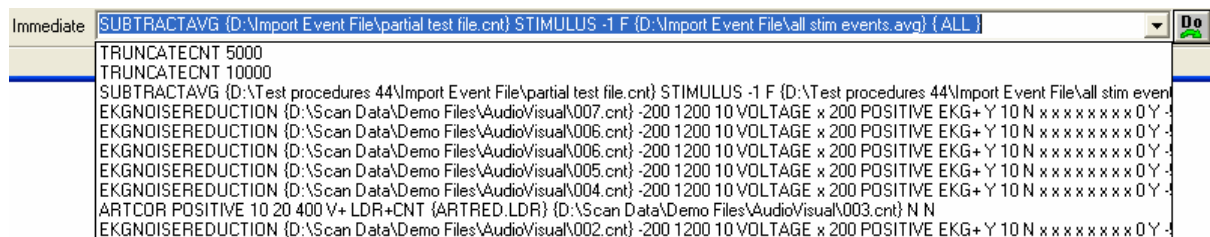
```


OPENFILE {C:\Scan Data\Demo Files\VisualAttention\viscpt.cnt}
CREATESORT SORT72
SORT72 -TypeEnabled T
SORT72 -TypeCriteria 1
EPOCH_EXPORT_INTERNAL "" N -100 996 N Y Y N N SORT72 {C:\Scan
Data\Demo Files\VisualAttention\viscpt.eeg}
DELETESORT SORT72
OPENFILE {C:\Scan Data\Demo Files\VisualAttention\viscpt.eeg}

```

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 field.




These can be run one line at a time by selecting the desired line (you must have the data file open first in order to apply a transform to it), and then clicking the  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.


3.3.1 ADD

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. You must have an AVG or EEG file open before using the command.

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

Example. ADD "c:\\Scan Data\\Batch Examples\\addfile.avg" "c:\\Scan Data\\newfile.avg".

3.3.2 ADDANNOTATION

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.

3.3.3 Appending Data Files

This section contains commands used to merge data files together.


3.3.3.1 APPEND

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:\Scan Data\secondfile.avg} {P3 P4 C3 C4} {c:\Scan Data\finalfile.avg}. Braces MUST be used with this command (not quotes; empty quotes will work). 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.


3.3.3.2 APPENDRECORDING

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:\Scan Data\file1.cnt} {c:\Scan Data\file2.cnt} {c:\Scan Data\file3.cnt}} {c:\Scan Data\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. (In this example, *file0.cnt* was already displayed, and the three files were appended to it to form the output file).


3.3.3.3 CONCATCNT

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

3.3.3.4 MERGE

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:\Scan Data\file1.eeg} {c:\Scan Data\file2.eeg} {c:\Scan Data\file3.eeg}} {c:\Scan Data\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. (In this example, *file0.eeg* was already displayed, and the three files were appended to it to form the 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
```

3.3.4 AREAREPORT

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.

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:\\Scan Data\\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 400ms. (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 example creates the DAT files for two data files.

```
set path "c:\\Scan Data\\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
```

3.3.5 Artifact Correction/Rejection

This section includes both the ocular artifact correction and the automatic artifact sweep rejection commands.

3.3.5.1 ARTCOR

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 the label of the blink channel. The maximum and individual sweeps will be reviewed, and the LDR and corrected CNT files are saved.

If you are using an EEG file instead of a CNT file, use the 8th parameter for the corrected output file path and name. This field will not be filled in automatically in the History or Immediate fields, since it is entered in a separate step when using the transform manually, at the end of the process.

3.3.5.2 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 μ Vs, 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).


3.3.5.3 ARTREJ_EX

ARTREJ_EX - Artifact Rejection, extended.  The ARTREJ_EX command is the same as the ARTREJ command, except that it allows you to specify in a list which channels should be the artifact rejection channels (10th parameter).

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 +/-50 μ Vs.

3.3.5.4 ARTREJCNT

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 had a somewhat different use in this case (see the EDIT manual for 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 (μ V)
6	double	Max amplitude (μ V)
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 $\pm 100\mu$ Vs, 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.

3.3.5.5 ARTREJCNTFREQ

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 (μ V)
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 μ Vs will result in a rejected block. Blocks will be recomputed using these criteria; Bad and Skipped channels are excluded.

3.3.5.6 ARTREJCNTTRIG

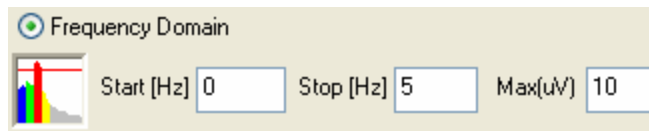
ARTREJCNTTRIG - Artifact Rejection for CNT files (using Triggers). ⚠ Regions around trigger events in CNT files can be rejected. A pre-trigger interval would be, for example, pre: -500 and post: -400. A post-trigger interval would be, for example, pre: 100 and post: 200. A symmetric interval about the trigger would be, for example, pre: -100 and post: 100.

1	double	Pre-trigger (ms)
2	double	Post-trigger (ms)
3	int	Trigger code
4	Boolean	Recompute

Example. ARTREJCNTTRIG -500 500 1 Y. In this example, the region bracketing type codes of 1, from -500 to 500ms, will be rejected. Saying Yes to the 4th parameter means that any previously rejected blocks will be accepted, and the new parameters will be applied (recomputed).

3.3.5.7 ARTREJFREQ

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 10 μ Vs (Bad and Skipped channels are excluded).

3.3.5.8 ARTREJFREQ_EX

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 (8th parameter).

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
8	list	List of artifact rejection channels

Example. `ARTREJFREQ_EX REJCRITERIA 0 5 Y 10 N N {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 10 μ Vs, using the indicated channels for artifact channels.

3.3.5.9 BLINKNOISEREDUCTION

BLINKNOISEREDUCTION - Blink Noise Reduction Routine. 🌀 This command provides a method for removing blinks that is the same as the online Blink Reduction correction. See also the `ARTCOR` command for an alternate (but similar) way to remove blink artifact (see the EDIT manual for a description of the differences). *You must have a Toolbox license to use this command.*

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 (μ Vs; 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 (μ V)
12	double	Artifact max (μ V)
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:\\Scan Data\\Demo Files\\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 μ V criterion. Artifact Rejection is enabled, using +/- 100 μ V thresholds. Events with type codes of 100 are placed at the trigger points. The bipolar reference and high pass filter options are not enabled.

3.3.5.10 CLEARART

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. 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.

3.3.5.11 EKGNOISEREDUCTION

EKGNOISEREDUCTION - EKG Noise Reduction. 🌀 The EKGNOISEREDUCTION command is used to reduce EKG and BCG artifact in CNT files. *You must have a license for the Toolbox in order to use the command.* The original command was also EKGNOISEREDUCTION. This was modified with the EKGNOISEREDUCTION_EX command, and then again with the EKGNOISEREDUCTION_EX2 command. For the V4.4 release, we are returning to the original EKGNOISEREDUCTION command. This will include all of the settings used in the V4.4 version of the software. If you have an existing batch file that uses a prior version of the EKGNOISEREDUCTION command (or _EX, or _EX2), you will need to replace it with the V4.4 version of the EKGNOISEREDUCTION command. EKGNOISEREDUCTION_EX and EKGNOISEREDUCTION_EX2 are obsolete and will not be recognized. If you have an existing batch file with the original EKGNOISEREDUCTION command, you will get an error message when you try to run it


with V4.4 (incorrect number of parameters). In other words, *you will need to replace any prior batch command for EKG Noise Reduction with the new version of the EKGNOISEREDUCTION command.*

Because of the large number of parameters used with this transform, it is recommended that you set the parameters in point & click mode, apply the transform, then copy and paste the line into the batch file you are creating.

1	string	Output file
2	double	Artifact Epoch Start
3	double	Artifact Epoch Stop
4	integer	Number of averages
5	defined value	Trigger type (QRS, VOLTAGE, RESPONSE, STIMULUS)
6	int	External Trigger Code
7	double	Voltage Threshold (μ V) (ignored if not using voltage triggering)
8	defined value	Trigger direction (POSITIVE, NEGATIVE)
9	string	Trigger channel label
10	Boolean	Insert Events
11	int	Insert Code
12	Boolean	Enable Trigger channel Filter
13	defined value	Filter Type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
14	float	High Pass cutoff
15	int	High Pass attenuation (db)
16	float	Low Pass cutoff
17	int	Low Pass attenuation (db)
18	float	Notch start frequency
19	float	Notch stop frequency
20	int	Notch attenuation (db)
21	double	Refractory period
22	Boolean	Enable Correlation
23	double	Correlation Interval Start
24	double	Correlation Interval Stop
25	int	Shift limit (data points)
26	double	Correlation threshold
27	string	Correlation channel label
28	Boolean	Enable Dilate average

EKGNOISEREDUCTION {D:\Scan data\output.cnt} -200 1200 10 VOLTAGE x 150 POSITIVE EKG+ Y 10 Y HIPASS 15 12 x x x x x 700 Y -50 50 25 95 EKG+ Y. The Epoch Interval was from -200 to 1200ms. 10 artifact sweeps were averaged. A voltage threshold was used for the triggering method, with a threshold of 150 μ V. The artifact was positive going in direction. EKG+ was the trigger channel. Event codes of 10 were inserted at the trigger points. A high pass filter was applied to the trigger channel, at 15Hz, 12dB. The refractory period was 700ms. Correlation was enabled, with an interval of +50ms, a Shift Limit of 25 points, and a correlation threshold of 95%, using EKG+ as the correlation channel. The Dilate average option was enabled.

3.3.5.12 FMRI

FMRI - fMRI Artifact Reduction.  The fMRI transform averages the MR slice artifacts (creating an average for each channel), and then subtracts the average from those sections containing the slices. The average start and stop points are determined by the placement of trigger events (see the EDIT manual for more

details). *The transform is included with the MagLink RT system, and you must have a MagLink RT license to use it* (contact techsup@neuroscan.com for details).

1	int	TR Trigger code
2	string	Trigger Method (VOLTAGE, STIMULUS, RESPONSE,)
3	double	TR start (ms)
4	double	TR duration (ms)
5	int	Number of slices per block
6	int	Number of Averages
7	defined value	Decimation Frequency (Hz)
8	Boolean	Enable Correlation
9	int	Shift limit (in sample points)
10	Boolean	Enable Continuous Removal
11	Boolean	Enable Filter
12	float	Low pass filter cutoff (Hz)
13	int	Offset (in points) to Voltage Threshold
14	float	Voltage Threshold
15	int	Preview channel number
16	string	Output file

Example. FMRI 1 STIMULUS 0.00 2000.00 32 10 1000 1 25 1 1 30.0 18591 5000.00 14 {D:\Data\output.cnt}. fMRI Reduction is performed using a stimulus event code of 1, TR Block with Start and End times of 0 and 2000ms, 10 averages, and a Decimated sampling rate of 1000Hz. Correlation is enabled, with a Shift limit of 25 points. Continuous Removal is enabled, and a 30Hz filter is applied. The trigger is 18,591 data points into the file. The voltage threshold was 5000, and the 14th channel (zero-based index) was selected as the Preview channel (again, these two parameters are ignored if you run the batch command).

3.3.5.13 READREJ

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:\\Scan Data\\Batch Examples\\p300.eeg"
REVIEW
SAVEREJ "c:\\Scan Data\\Batch Examples\\rejsweeps.rej"
CLEARART
REVIEW
READREJ "c:\\Scan Data\\Batch Examples\\rejsweeps.rej"

```

In this example, the *p300.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.

3.3.5.14 SAVEREJ

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:\\Scan Data\\Batch Examples\\rejectsweeps.rej".

3.3.6 Averaging

The primary command used to average single sweeps of either time or frequency domain files is the AVERAGE command, described below. The GROUPAVG command is used to average AVG files to form a group average.

3.3.6.1 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 CREATSORT). 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:\\Scan Data\\Batch Examples\\averagedfile.avg". This will create an average of the sweeps using the MySort file (created beforehand with the CREATSORT command).

3.3.6.2 AVERAGE_EX

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:\Scan Data\output file.avg}. Sweeps from a previously retrieved epoched file are averaged in the time domain, using the prestimulus and poststimulus intervals for SNR estimation, with no sorting.

3.3.6.3 GROUPAVG

GROUPAVG - Group File Average. ~ This command is used to combine AVG files to form a group averaged AVG file. *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:\Scan Data\Batch Examples\file1.avg} {c:\Scan Data\Batch Examples\file2.avg} {c:\Scan Data\Batch Examples\file3.avg}} G Y 1 N N {c:\Scan Data\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).

3.3.7 AVGBANDS


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:\\Scan Data\\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 more complete example might look like the following:

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

3.3.8 Baseline Correction

Baseline Correction.  The current command for performing Baseline Correction is BASECOR_EX2, described below.

3.3.8.1 BASECOR

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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\file1.eeg"
BASECOR PRE 0 0 N N "$path\\basefile1.eeg"
CLOSEFILE "file1.eeg"
```

For AVG files:

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

3.3.8.2 BASECOR_EX

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").

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

3.3.8.3 BASECOR_EX2

BASECOR_EX2 - Exclude Bad or Skipped Channel(s) for Baseline Correction. This command differs from the BASECOR_EX command in that it allows you to exclude Bad or Skipped channels.

1	defined value	Interval type (FIRSTPOINT, PRESTIMINTERVAL, ENTIREINTERVAL, USERDEFINED)
2	double	Interval start latency (only if 1 is USERDEFINED)
3	double	Interval stop latency (only if 1 is USERDEFINED)
4	Boolean	Exclude Bad channels
5	Boolean	Exclude Skipped channels
6	list	List of electrodes to correct, or "All"
7	string	Output file (not needed for AVG files)

Example. BASECOR_EX2 PRESTIM x x Y Y {ALL} {}. Baseline correction will be applied using the prestimulus interval for all channels, excluding any Bad or Skipped channels (no output file is needed for AVG files).

3.3.8.4 EXCLUDEFORBASECOR

EXCLUDEFORBASECOR - Exclude Channel(s) for Baseline Correction. This command is generally not needed with the BASECOR_EX2 command.

1...n list Electrode name(s)

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


3.3.8.5 RESETFORBASECOR

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. This command is generally not needed with the BASECOR_EX2

command.


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

3.3.9 CLEARALLMARKERS

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.


3.3.10 CLEARMARKERS

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.

3.3.11 CLOSEALL


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.

3.3.12 CLOSEFILE


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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\file 1.avg"
OPENFILE "$path\\file 2.avg"
ARRANGEWINDOWS TILEV
PAUSE
CLOSEFILE "file 1.avg"
```


3.3.13 COHERENCE

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:\\Scan Data\\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).

3.3.14 COHERENCE_EX

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


3.3.14.1 SETCOHREF

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.


3.3.15 COMPAREELECTRODES

COMPAREELECTRODES - Compare Electrodes.  This command allows you to compare up to 8 electrodes in a Multiple Window display. (See also REMOVECOMPARE-ELECTRODES).

1	string	Primary Electrode name
2	string	List of Compare Electrode name(s) or All.

Example. COMPAREELECTRODES OZ {PZ PO1 PO2 O1 O2}. In this example, the electrodes in the list will be displayed with the OZ electrode.

3.3.16 CORRELATESPEAKS

CORRELATESPEAKS - Correlate Peaks.  The CORRELATEPEAKS command was taken from the EKG Noise Reduction transform. It performs a series of intra-class correlations in order to align peak events by waveform similarity rather than event code placement (see the EDIT and MagLink RT manual for more details).

1	defined value	Trigger type (STIMULUS, RESPONSE)
2	int	Type code
3	double	Correlation interval Start (ms)
4	double	Correlation interval End (ms)
5	double	Shift Limit (points)
6	int	Averages
7	string	Correlation Channel label
8	defined value	Bad peaks (PROCESS, REMOVE, CHANGECODE)
9	double	New event code (ignored if 8 is PROCESS or REMOVE)
10	double	Correlation Threshold (ignored if 8 is PROCESS)

Example. CORRELATEPEAKS STIMULUS 10 -50 50 10 10 EKG CHANGECODE 20 95. A stimulus type code of 10 is used to define the peak of the blinks. The correlation interval is -50 to 50ms, with a Shift Limit of 10 points. 10 sweeps are averaged to create the average artifact. Sweeps with correlations less than 95% are excluded from the average artifact, and given a new type code of 20.

3.3.17 CREATESORT/DELETESORT

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 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)	CORRECT NORESPONSE)
-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).


3.3.18 CUTEPOCH

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 or equal to 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 "\\Scan Data\\Batch Examples\\outputfile.avg". In this example, a new file (.avg or .eeg file) from -100 to 500ms will be created. (Start and End latencies will be adjusted automatically to the nearest actual data point, if required).

3.3.19 DCCORRECT

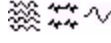
DCCORRECT - DC Correction.  The DC Correction option will attempt to correct for DC offsets and DC drifting in continuous files recorded on a SynAmps in DC mode with a DC High Pass filter (see the EDIT manual for complete details).

1	double	Precorrection start time
2	double	Precorrection stop time
3	double	Post correction start time
4	double	Postcorrection 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 N N -200 996 "" 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.


3.3.20 DEBLOCK

DEBLOCK - Deblock Artifact.  The DEBLOCK option is used to replace stimulus artifact with a flat line. Windowing is performed to reduce any abrupt transitions at the beginning and end of the lines.

1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Taper percentage
3	int	Trigger code (CNT files only)
4	double	Start latency (EEG and AVG time domain files only)
5	double	End latency (EEG and AVG time domain files only)
6	list	Electrode name(s) to correct, or "All"
7	string	Output file (CNT and EEG files only)

Example. DEBLOCK COSINE 10 2 0 10 { ALL } {C:\Scan Data\Demo Files\Visual-Attention\deblock.cnt}. Deblocking will be applied using a Cosine Window type with a 10% taper, to all channels, wherever a type code of 2 occurs, and with a span from 0 to 10ms duration. With AVG files, enter {} for the final parameter, or else leave it blank.

3.3.21 DECIMATE

DECIMATE - Decimation.  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 can be applied (passed four times) to correct for aliasing.

1	int	New AD Rate
2	Boolean	Enable Filter
3	float	Low pass value
4	int	Slope (poles; 0=6dB/oct, 1=12dB/oct, 2=24dB/oct, 3=48dB/oct)
5	string	Output file

Example. DECIMATE 250 1 30.0000 2 {D:\Scan Data\output file.cnt}. The new CNT file will be created with an AD rate of 250Hz, and a 30Hz 24dB/oct filter will be applied.

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.


3.3.22 DELETEBADCHANNELS

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:\\Scan Data\\Batch Examples\\NoBadChannels.cnt".


3.3.23 DELETESWEEPS

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:\\Scan Data\\Batch Examples\\NoRejSweeps.eeg".

3.3.24 Detrending (Linear Detrending)

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

3.3.24.1 DETREND

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:\\Scan Data\\Batch Examples\\detrended file.eeg". The channels will be detrended using the entire interval, and a new file is created.

3.3.24.2 DETREND_EX

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").

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 (or "All")
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:\\Scan Data\\Demo\\Veps\\output.eeg}. Linear detrending will be applied using the prestimulus interval for the selected channels.

3.3.24.3 EXCLUDEFORDETREND

EXCLUDEFORDETREND - Exclude Channel(s) for Linear Detrend. Use this command to exclude channels from Detrending. You will generally not need to use this command if you are using DETREND_EX.

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

Example. EXCLUDEFORDETREND {Fp1 Fp2 F7 F8}.

3.3.24.4 RESETFORDETREND

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 detrending. It avoids having to reset each channel individually. You will generally not need to use this command if you are using DETREND_EX.

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

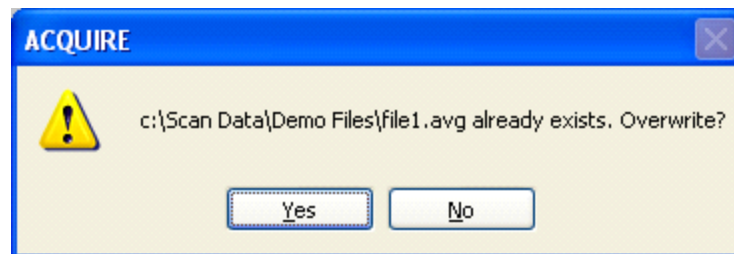
3.3.25 ENABLEOVERWRITEPROMPT

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:\\Scan Data\\Demo Files"
OPENFILE "$path\\file1.avg"
BASECOR_EX2 PRESTIMINTERVAL x x N N { ALL } {}
SAVEAS "$path\\file1.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:\\Scan Data\\Demo Files"
OPENFILE "$path\\file1.avg"
BASECOR_EX2 PRESTIMINTERVAL x x N N { ALL } {}
ENABLEOVERWRITEPROMPT N
SAVEAS "$path\\file1.avg"
```



Note

You cannot overwrite an open CNT file, as was done in the example above with AVG files. You will need to save the CNT file with a different file name.


3.3.26 EPOCH

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 **CREATESORT** command. (It may be necessary to merge the DAT file - **MERGE EVT** - from STIM beforehand to have access to the behavioral information).


3.3.27 EPOCH_EX

EPOCH_EX - Epoch, Extended.  The EPOCH_EX command differs from the original EPOCH command in that it will accept an event file that has 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_EX 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 **CREATESORT** command. (It may be necessary to merge the DAT file - **MERGE EVT** - from STIM beforehand to have access to the behavioral information).

3.3.28 ERBP

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.

3.3.29 ERCOH

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.

3.3.30 Exporting/Importing ASCII/EDF Files

These commands are used for exporting files to and importing files from ASCII or EDF.

3.3.30.1 EXPORTAVG

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

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

Example. EXPORTAVG "c:\\Scan Data\\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:\\Scan Data\\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
}

```


3.3.30.2 EXPORTAVG_EX

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 12th parameter will append the new data to an existing data file with the same name (if the file already exists). The last parameter lets you select the channels to be exported.

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
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:\Scan Data\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.


3.3.30.3 EXPORTAVG_EX2

EXPORTAVG_EX2 - Export AVG files, Extended.  Comma Delimiting has been added to the EXPORTAVG_EX command (13th parameter). Comma delimiting separates the data points by commas, rather than by tabs. The parameters for EXPORTAVG_EX2 are:

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
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	Boolean	Use comma delimiting
14	list	Channels to be exported (or "all")

Example. EXPORTAVG_EX2 {C:\Scan Data\Demo\output.dat} P Y Y Y Y N Y Y Y N 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). Comma delimiting has been disabled. All channels will be exported.

3.3.30.4 EXPORTCNT

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
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:\Scan Data\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.

3.3.30.5 EXPORTCNT_EX


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:\\Scan Data\\Batch Examples\\cntfile.dat" Y Y 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.


3.3.30.6 EXPORTCNT_EX2

EXPORTCNT_EX2 - Export CNT File to ASCII, Extended.  Comma Delimiting has been added to the EXPORTCNT_EX command (11th parameter). Comma delimiting separates the data points by commas, rather than by tabs.

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	Boolean	Use comma delimiting
12	list	Channels to be exported (or "all")

Example. EXPORTCNT_EX2 "c:\\Scan Data\\Batch Examples\\cntfile.dat" 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. The data labels are included and the values are written with maximum resolution (to at least eight decimal places). Comma delimiting has not been enabled. All channels will be exported.


3.3.30.7 EXPORTCOH_EX

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
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:\\Scan Data\\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.

3.3.30.8 EXPORTCOH_EX2

EXPORTCOH_EX2 - Export COH files, Extended.  Comma Delimiting has been added to the EXPORTCOH_EX command. Comma delimiting separates the data points by commas, rather than by tabs.

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

Example. EXPORTCOH_EX2 "c:\\Scan Data\\Batch Examples\\cohfile.dat" PAIRS Y N 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. Comma delimiting has not been enabled.


3.3.30.9 EXPORTEDF

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
2	Boolean	Include bad channels
3	Boolean	Include skipped channels
4	list	Channels to be exported

Example. EXPORTEDF "c:\\Scan Data\\demo\\EDF file.edf" Y Y {All}. The CNT file is exported in EDF, with all channels exported, including the bad and skipped channels.

3.3.30.10 EXPORTEDF_EX

EXPORTEDF_EX - Export 32 bit data to EDF.  The EXPORTEDF command was modified to provide various scaling methods that can be used when exporting 32 bit data files to the 16 bit EDF format (see the EDIT manual for details).

1	string	Output file
2	Boolean	Include Bad channels
3	Boolean	Include Skipped channels
4	list	Channels to be exported
5	defined value	Scaling method (AMPLIFIERRESOLUTION, PERCHANASYMMETRICAL, PERCHANASYMMETRICAL, ALLCHANASYMMETRICAL, ALLCHANASYMMETRICAL, USER)
6	double	User scaling Min
7	double	User scaling Max
8	defined value	Block size (FIXED, AUTO)
9	double	Block seconds

Example. EXPORTEDF_EX {D:\Scan Data\Demo Files\EEGs\sample.edf} F T { ALL } PERCHANASYMMETRICAL x x AUTO x. All channels in the file are exported with Bad channels excluded, Skipped channels included, using the Per Channel Asymmetrical method with Automatically determined block size.



3.3.30.11 EXPORTEEG

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

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

Example. EXPORTEEG "c:\Scan Data\Batch Examples\leegfile.dat" P Y Y 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.

3.3.30.12 EXPORTEEG_EX

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: [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. 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	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:\\Scan Data\\Batch Examples\\eegfile.dat" P Y Y 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.


3.3.30.13 EXPORTEEG_EX2

EXPORTEEG_EX2 - Export EEG File to ASCII, Extended. *** Comma Delimiting has been added to the EXPORTCOH_EX command (13th parameter). Comma delimiting separates the data points by commas, rather than by tabs.

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
10	Boolean	Include data labels
11	Boolean	Use maximum resolution
12	Boolean	Append if file exists
13	Boolean	Use comma delimiting
14	list	Channels to be exported (or "all")

Example. EXPORTEEG_EX2 "c:\\Scan Data\\Batch Examples\\eegfile.dat" P Y Y N N N Y Y N 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). Comma delimiting has not been enabled. All channels will be exported.


3.3.30.14 IMPORTAVG

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
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:\\Scan Data\\Batch Examples\\avgascii.dat" P 30 1000 -50 550 200 N. The ASCII data will be imported.


3.3.30.15 IMPORTCNT

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
2	int	Number of channels
3	double	Acquisition rate
4(opt)	defined value	Company (NEUROSCAN, NK)

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


3.3.30.16 IMPORTEEG

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
2	defined value	Method (POINTS, ELECTRODES)
3	int	Number of channels
4	double	Acquisition rate
5	double	X min (in seconds)
6	int	Number of points
7	int	Number of sweeps
8	Boolean	Frequency domain

Example. IMPORTEEG "c:\\Scan Data\\Batch Examples\\eegascii.dat" P 30 1000 -.050 550 200 N. The ASCII data will be imported.


3.3.30.17 IMPORTEEG_EX

IMPORTEEG_EX - Import EEG File from ASCII.  In the prior IMPORTEEG batch command, X Minimum was specified in *seconds*. A new batch command has been added: IMPORTEEG_EX. The parameters are the same as before, only with this command X Minimum is in *milliseconds*.

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

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


3.3.31 EXTRACT

EXTRACT - Extract Channels.  The EXTRACT command creates a file containing data from selected channels from the currently open 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:\\Scan Data\\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.

3.3.32 EXTRACTBLOCK

EXTRACTBLOCK - Extract Block from CNT file.  This command is used to create new files from blocks of data extracted from CNT files. Blocks can be defined by stimulus or response events, or time points. Multiple output files may be created, as described below. Each will have a suffix added automatically (*_1.cnt, *_2.cnt, *_3.cnt etc.).

1	defined value	Method (STIMCODE, RESPONSECODE, INTERVAL, TIME)
2	integer	Event Code (STIMCODE and RESPONSECODE only)
3	double	Interval (ms) (INTERVAL only)
4	double	Start time (ms) (TIME only)
5	double	End time (ms) (TIME only)
6	string	Output file (suffixes may be appended)

Example. EXTRACTBLOCK STIMCODE 55 x x x "d:\\data files\\stimcode.cnt". In this file there were two type codes of 55. Three output files were created: stimcode_1.cnt (beginning of file to first event), stimcode_2.cnt (first event to second event), and stimcode_3.cnt (second event to end of file).

One or more output files will be created depending on the Method, number of events, or duration of the Interval you specify.




STIMCODE/RESPONSECODE. If you have a single stimulus or response event code in the file, two cnt files will be created. The first is from the beginning of the file to the event, and the second is from the event to the end of the file. With two events, there are three output files, and so on.

INTERVAL. If you enter an INTERVAL of, for example, 20,000ms, there will be multiple cnt files created with consecutive 20 seconds blocks. The last block will contain whatever time span remains (it may be less than 20 seconds).

TIME. In this mode only a single output file will be created, with no added suffix. If you enter, for example, 5000 and 10000ms, the new file will contain that block only.

Entering stimulus events manually. As a reminder, you can enter a *stimulus* event manually at any point in a CNT file by positioning the cursor, pressing *Ctrl+V*, and entering a code (1-255).

3.3.33 Filtering data files

Filtering data files.    The principle command for filtering data files is FILTER_EX.

3.3.33.1 FILTER

FILTER - Filter. Use this command to perform the following 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:\\Scan Data\\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.

3.3.33.2 FILTER_EX

FILTER_EX - Filter, Modified. This command differs from the FILTER command in that you may select the filter class (FIR or IIR). It also allows you to list the electrodes that you want to include (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.

3.3.33.3 EXCLUDEFORFILTER

EXCLUDEFORFILTER - Exclude Channel(s) for Filtering. Use this command to exclude channels from Filtering. The command is generally not needed if you are using the newer FILTER_EX command.

1...n list Electrode name(s)

Example. EXCLUDEFORFILTER {Fp1 Fp2 F7 F8}.

3.3.33.4 RESETFORFILTER

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

not needed if you are using the newer FILTER_EX command.

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

3.3.34 FSP

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 CREATSORT.

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:\\Scan Data\\Batch Examples\\Fspoutput.avg". This line performs a basic FSP calculation, with no sorting.

3.3.35 FSP_EX

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").

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:\\Scan Data\\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.

3.3.36 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).

3.3.36.1 GETCOMP/DELETEALLCOMP

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:\\Scan Data\\Batch Examples\\myfile.avg".

3.3.36.2 GETCURRENTEPOCHINDEX/SETCURRENTEPOCHINDEX

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.

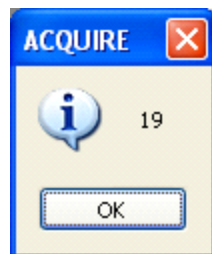
Return. The sweep number of the currently displayed epoch is returned.

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).

3.3.36.3 GETEPOCHCOUNT

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.



3.3.36.4 GETEPOCHINFO / SETEPOCHINFO

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)

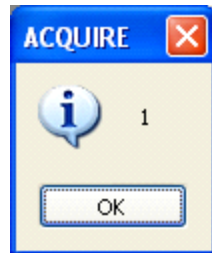
Example. GETEPOCHINFO 45 -A. The program will return the "Accept" field for the 46th sweep. (Reminder: with a zero-based index, the first sweep is really sweep 0, although it is displayed as sweep 1 in the Status Box). To see the returned value, use a script such as the following:

```

set path "c:\\Scan Data\\Batch Examples"
OPENFILE "$path\\viscpt.eeg"
set accept [GETEPOCHINFO 45 -A]
INSTRUCT $accept
CLOSEALL

```

The returned value will appear as shown.



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:\\Scan Data\\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.

3.3.36.5 GETEVENTCOUNT

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.



3.3.36.6 GETEVENTINFO/SETEVENTINFO

GETEVENTINFO / SETEVENTINFO - Get/Set Event Information.  These commands 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)

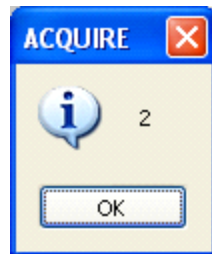
Depending upon which Parameters you selected for #2, the program will return different information. The following translates the meaning of the returned codes.

-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:\\Scan Data\\Batch Examples\\viscpt.cnt"
set number [GETEVENTINFO 123 -Stim]
INSTRUCT $number
```

This returns the value shown.



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:\\Scan Data\\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:\Scan Data\Batch Examples\AllEvents.cnt}
ARRANGEWINDOWS TILEH
set eventcount [GETEVENTCOUNT]
SAVEEVENT Y Y Y Y Y Y {c:\Scan Data\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:\Scan Data\Batch Examples\After.EV2}

```

3.3.36.7 GETMARKERCOUNT

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)

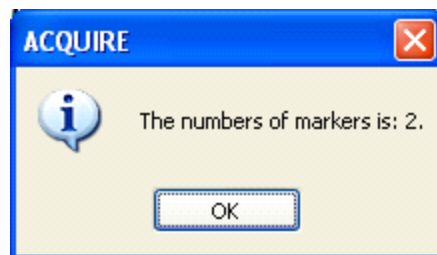
Example. GETMARKERCOUNT "OZ" 50. The number of markers in the 50th sweep (zero-based) at the OZ channel is returned. A sample sequence is as follows:

```


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.



3.3.36.8 GETMARKERINFO

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."

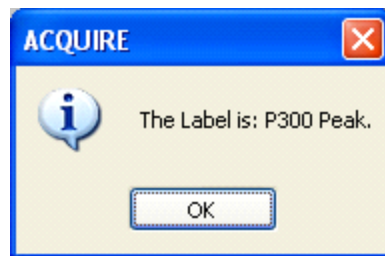
```

```
set label [GETMARKERINFO "OZ" 0 LABEL 50]
INSTRUCT "The Label is: $label."
```


The first value returned is:



The second returned value is:



3.3.36.9 GETPOINTDATA

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:

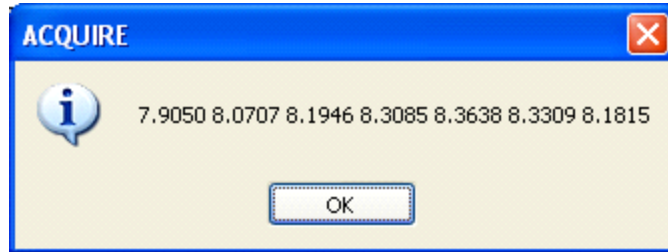
1	int	Start points (zero-based index)
2	int	End points (zero-based index)
3	string	Electrode label
4 (EEG files only)	int	Epoch number (zero-based index)

Return. The data point values within the specified range and epoch are returned.

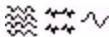
Example. GETPOINTDATA 130 170 "OZ". The 130-170th data points, from the OZ electrode, are displayed. The last argument was omitted (an AVG file was opened, rather than an EEG file). If you know the AD rate and the starting point of the sweep, you can calculate the Start and End index numbers, or, you 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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\vcp.avg"
set start [LATENCYTOPOINT 97]
set stop [LATENCYTOPOINT 104]
```

```
set data [GETPOINTDATA $start $stop {OZ}]
INSTRUCT $data
CLOSEALL
```




3.3.37 GFP

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:\\Scan Data\\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.


3.3.38 IMPORTEVENT

IMPORTEVENT - Import Event File.  This command is used to import an event file (.evt or .ev2) into a CNT file. The event file can replace or be added to the existing event table, and the events can be in byte offsets or second offsets.

1	Boolean	Replace events
2	Boolean	Use seconds
3	string	Event file name

Example. IMPORTEVENT N N {D:\\Scan Data\\Demo Files\\VisualAttention\\event file.ev2}. An event file is imported without replacing the existing event table. The offsets are in ms, and the selected file is shown.

3.3.39 INSERTRESPONSEEVENT

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

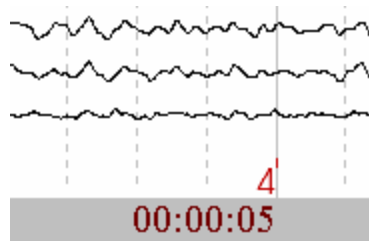
Example. INSERTRESPONSEEVENT 2500 4. This enters a stimulus type code of 4 at the 2,500th data point.

The script may appear similar to the following, in which the event is inserted, and an event file is created:

```
set path "c:\\Scan Data\\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.



3.3.40 INSERTSTIMEVENT

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:

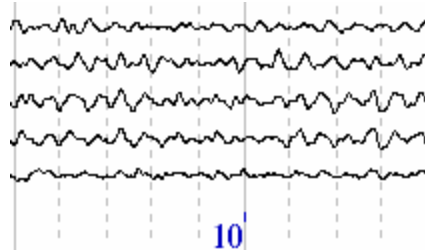
```
set path "c:\\Scan Data\\Batch Examples"
```

```

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

```

Note: when you add an event to the event table, the event is immediately placed, and therefore the size of the event table has grown and the index of the current and all subsequent events is incremented (i.e., the list is resorted each time an event is added). Since this could result in confusion when multiple events are inserted, it is recommended that you build a new list of events as you go through the event table, and then go through again using the new list and insert the new events.

For example, let's say you want to enter a specified stimulus event, at a specified time prior to an existing specified stimulus event. In other words, you want to insert a stimulus type code of 20, 500ms prior to an existing stimulus type code of 10. Examine the following batch file.

```

# Initially, you select the CNT file and enter the desired type code and time value.
set file [GETINPUTFILE "Select file" "cnt"]
set path [PICKOFFDIRECTORY $file]
set filename [PICKOFFNAMEONLY $file]
set suffix "_added-events"
set ext ".cnt"
OPENFILE "$file"
set totalevents [GETEVENTCOUNT]
set stimcode [GETINPUT "LOOK FOR STIM CODE:"]
set newstimcode [GETINPUT "INSERT STIM CODE:"]
set time [GETINPUT "ENTER TIME IN MS"]

#The sample rate is obtained by getting the point number at one second (1000ms)
set samprate [LATENCYTOPOINT 1000]

set points [expr $time*$samprate/1000]

```

```

set NewEventList {} ;#create an empty list

#Determine where the new events should be added.
#Events are not actually added yet because this would cause the size of the event list to
change.
#So we simply store the offsets in a list


for {set i 0} {$i < $totalevents} {incr i} {
    set event[GETEVENTINFO $i-EventType]
    if {$event == "STIMULUS"} {
        set code[GETEVENTINFO $i-StimulusCode]
        if {$code == "$stimcode"} {
            set offset[GETEVENTINFO $i-Offset]
            set newoffset[expr $offset-$points]
            lappend NewEventList $newoffset
        }
    }
}

#Now add all the new events into the list
foreach element $NewEventList {
    INSERTSTIMEVENT $element $newstimcode 00 NORESP
}

SAVEAS "$path$filename$suffix$ext"
CLOSEALL

```


3.3.41 INTEGRATEWAVEFORM

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 (or "")

Example. INTEGRATEWAVEFORM N "\$path\\output file.avg". The integration is performed without rectifying the waveforms, and the output file is saved.

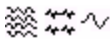
3.3.42 LDR

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:\\Scan Data\\Batch Examples\\bipolar.ldr" "c:\\Scan Data\\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 - μ V, 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.


3.3.43 LOAD3DD

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:\\Scan Data\\position.3dd}. The indicated 3DD file is loaded to a previously retrieved data file.

3.3.44 MERGEEVT

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.

1	string	Task file name
---	--------	----------------

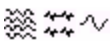
Example. MERGEEVT "c:\\Scan Data\\Batch Examples\\stimfile.dat". The full BATCH sequence might appear as follows (with the "path" variable substitution):

```
set path "c:\\Scan Data\\Batch Examples"
OPENFILE "$path\\viscpt.cnt"
MERGEEVT "$path\\viscpt.dat"
SAVEAS "$path\\newviscpt.cnt"
```

3.3.45 Open/Save Commands

In most instances, you should use the OPENFILE and SAVEAS_EX commands.

3.3.45.1 OPENFILE

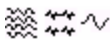
OPENFILE - Open File.  This command is used to open all types of data files in the EDIT program. As a general rule, you should use OPENFILE after every file you create. For example, if you output an ocular artifact corrected file, use OPENFILE to retrieve it before applying any other commands (otherwise 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:\\Scan Data\\data\\myfile.eeg". The indicated file will be opened.

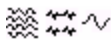
3.3.45.2 SAVEAS

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:\\Scan Data\\Batch Examples\\newfile.cnt". The indicated file will be saved to the designated folder.


3.3.45.3 SAVEAS_EX

SAVEAS_EX - Save File As, with DC option.  This is the same as the SAVEAS command, with the additional option to save DC offset and drift corrections permanently (following application of the DC Offset Correction transform; CNT files only).

1	string	Output file name (or "")
2	Boolean	Make DC correction permanent

Example. SAVEAS "c:\\Scan Data\\Batch Examples\\newfile.cnt" Y. The indicated file will be saved to the designated folder, with the DC corrections.

3.3.46 Peak Detection

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 SETSPECTDISPLAYSTYLE 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_EX command.

3.3.46.1 EXCLUDEFORPEAKDETECTION

EXCLUDEFORPEAKDETECTION - Exclude Channel(s) for Peak Detection. Use this command to exclude channels from Peak Detection. This command is generally not needed if you are using the newer PEAKDETECTION_EX command.

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

Example. EXCLUDEFORPEAKDETECTION {Fp1 Fp2 F7 F8}.

3.3.46.2 PEAKDETECTION

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	float	Start search point (ignored if param 2 is False)
4	float	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).

3.3.46.3 PEAKDETECTION_EX

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 peak detection (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.

3.3.46.4 RESETFORPEAKDETECTION

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. This command is generally not needed if you are using the newer PEAKDETECTION_EX command.

Example. The RESETFORPEAKDETECTION command is written on a line by itself, and uses no parameters.

3.3.46.5 SAVEPEAK

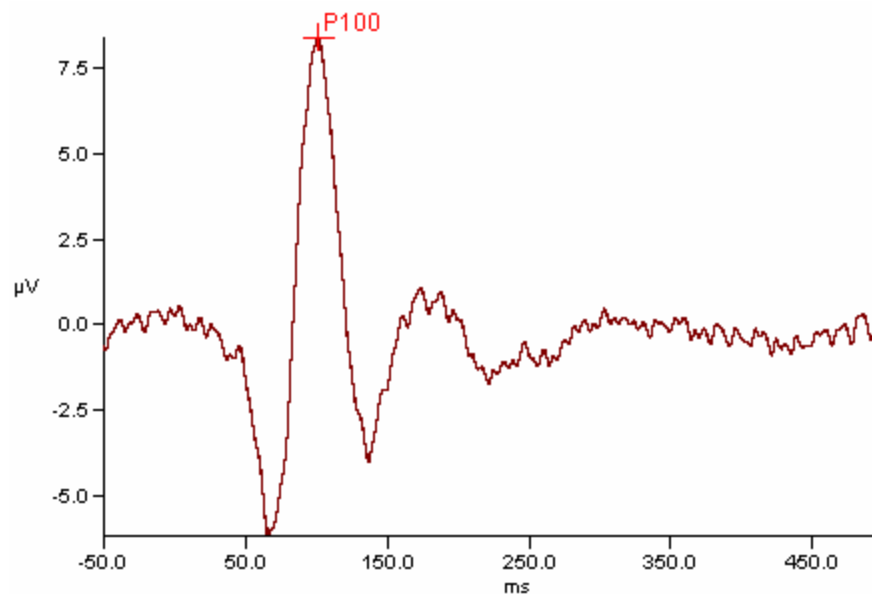
SAVEPEAK - Save Peak File. The SAVEPEAK command is used to create/open an ASCII file (DAT extension), and to save the peak detection information to it. If you retrieve an existing DAT file, the new peak information will be added to the end of the file. If the -SubstituteBad option is selected, then the 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 "")
2(optional)	defined value	Option (-SubstituteBad)
3(optional)	string	Substitution string

Example. SAVEPEAK "c:\\Scan Data\\Batch Examples\\peakdata.dat"
-SubstituteBad "Bad Electrodes". Below is a sequence of commands that includes saving the DAT file.

```
set path "c:\\Scan Data\\Demo Files\\Veps"
OPENFILE "$path\\vcp.avg"
PEAKDETECTION_EX "P100" Y 80 120 MAX Y "OZ" Y Y USER RED
{ ALL }
SAVEPEAK "$path\\peakdata.dat" -Sub "Bad Electrodes"
```

The labeled 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	01	P100		0.101000	5.798689
1	02	P100		0.101000	6.215950

3.3.47 PRINT

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.


3.3.48 QRS

QRS - QRS Detection.  This command will perform an automated detection of QRS complexes and insert a stimulus type code for each.

1	string	Trigger Channel
2	int	Stim code

Example. QRS EKG+ 5. QRS detection is performed using the EKG+ channel. A stimulus event code of 5 is inserted for each event.


3.3.49 RECTIFY

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:\\Scan Data\\Batch Examples\\rectfile.eeg". The file in working memory is rectified and the results are written to a new file.


3.3.50 RECTIFY_EX

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:\\Scan Data\\Batch Examples\\rectfile.eeg". All channels of the file in working memory are rectified, and the results are written to a new file.

3.3.51 REFER

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 are 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.

3.3.52 REJECTBLOCK

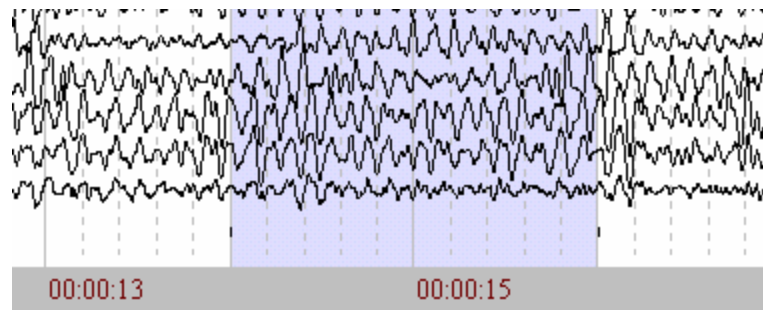
REJECTBLOCK - Reject Block.  Use this command to reject blocks of data in CNT files. (You will need the AD rate to compute the offset: $\text{offset} = \text{AD}/1000 \times \text{ms}$).

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:\\Scan Data\\Batch Examples"
OPENFILE "$path\\closed.cnt"
REJECTBLOCK 7000 8000
```

The section between the 14-16 seconds is rejected (AD = 500).




3.3.53 REMOVECOMPAREELECTRODES

REMOVECOMPAREELECTRODES - Remove Compared Electrodes.  This command removes all or selected electrodes that have been compared with the COMPARE-ELECTRODES command.

1	string	Primary Electrode name
2	string	Compare Electrode name(s) or All to be removed

Example. REMOVECOMPAREELECTRODES OZ {PZ PO1 PO2}. The compared electrodes in the list are removed from the OZ Multiple Window display.

3.3.54 REMOVEEVENT

REMOVEEVENT - Remove Event from CNT File.  This command is used to remove events from CNT files, one event at a time.

1 int Index of event (zero-based)


Example. REMOVEEVENT 25. The 25th event in the CNT file will be removed (the first event is 0). This command may be used effectively within a loop. The following example is the **Stim2resp.tcl** program, which converts responses made with the mouse and keyboard (blue) to those made by the response pad (red).

```
set stim [GETINPUT "Stimulus code to change" "Stimulus code"]
set resp [GETINPUT "Response code to substitute" "Response code"]
set evcount [GETEVENTCOUNT]

for {set i 0} {$i < $evcount} {incr i} {
    set type [GETEVENTINFO $i -EventType]
    set offset [GETEVENTINFO $i -Offset]
    set code [GETEVENTINFO $i -StimulusCode]

    if {$type == "STIMULUS" && $code == $stim} {
        REMOVEEVENT $i
        incr i -1
        INSERTRESPONSEEVENT $offset $resp
    }
}
```

3.3.55 RESPWIN


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
7	string	<i>Descriptor</i>
8	string	<i>Target vector (1, 2, 3-6, etc.)</i>
9	int	<i>Target correct response</i>
10	string	<i>Distractor vector (1, 2, 3-6, etc.)</i>
11	int	<i>Distractor correct response (or 0)</i>

Repeat parameters 7-11 for each target/distractor pair.

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).


3.3.56 SAVEEVENT

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 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.

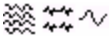
3.3.57 SAVEEVENT_EX

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 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.

3.3.58 SELECTFILE


SELECTFILE - Select Open Data File.  This command allows you to switch the "focus" 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:\\Scan Data\\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
```

3.3.59 Smoothing Data

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_EX2.

3.3.59.1 EXCLUDEFORSMOOTH

EXCLUDEFORSMOOTH - Exclude Channel(s) for Smoothing. Use this command to exclude channels from Smoothing. This command is generally not needed

if you are using the newer **SMOOTH_EX** command.

1...n list Electrode name(s)

Example. EXCLUDEFORSMOOTH {Fp1 Fp2 F7 F8}.

3.3.59.2 RESETFORSMOOTH

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. This command is generally not needed if you are using the newer **SMOOTH_EX** command.

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

3.3.59.3 SMOOTH

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:\\Scan Data\\Batch Examples\\smoothed data.eeg"**.
Three adjacent data points are smoothed, and three passes are made.

3.3.59.4 SMOOTH_EX

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"). 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:\\Scan Data\\demo\\output file.eeg"**. The indicated electrodes will be smoothed.


3.3.59.5 SMOOTH_EX2

SMOOTH_EX2 - Display Dialog for Smooth. This command differs from the SMOOTH_EX command in that it allows you to display the Smoothing dialog window before the transform is applied. 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	Boolean	Show dialog
5(optional)	string	EEG output file name

Example. SMOOTH_EX 3 3 {C3 CZ C4} Y "c:\\Scan Data\\demo\\output file.eeg}. The Smoothing dialog screen will be displayed, allowing you to set the parameters before the transform is applied.


3.3.60 SNR

SNR - 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 to 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 see the SNR values, *right click* between electrode windows and select **View SNR Values**. To view the SNR values later, save the file with the modifications.

3.3.61 SORT

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
---	--------	-----------

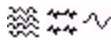
2 string Output file name (or "")

Example. SORT "sortfile" "c:\\Scan Data\\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:\\Scan Data\\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"
DELETESORT sortfile
OPENFILE "$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.

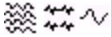
3.3.62 SPATIALFILTER

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
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
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 0 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.

3.3.63 SPATIALSVD

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.

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:\\Scan Data\\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"
```

3.3.64 Spectral analyses

Spectral analyses. Average waveforms consisting of a power-of-2 number of points (e.g., 64, 128, 256, 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 **SPLINEFIT** command 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.

3.3.64.1 FORWARDFFT/INVERSEFFT

FORWARDFFT/INVERSEFFT - Forward FFT/Inverse 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 (or "")

Example. FORWARDFFT "c:\\Scan Data\\Batch examples\\forwardFFT.eeg".
 INVERSEFFT "c:\\Scan Data\\Batch Examples\\originaldata.eeg".

Below is an example script using FORWARDFFT and INVERSEFFT using the SPLINE command.

```
set path "c:\\Scan Data\\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"

# CLOSSES 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

# CLOSSES ALL OPEN DATA WINDOWS
CLOSEALL
```


3.3.64.2 SPECTRUM

SPECTRUM - Spectrum.  The SPECTRUM command will compute a Fast Fourier Transform on an AVG file. The points in the AVG file must be a power of 2 (256, 512, 1024, etc.). If not, use the SPLINEFIT command to recompute with a power of 2.

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:\\Scan Data\\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.

3.3.65 SPLINEFIT

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. In general, you should always spline fit down to the next power of 2; avoid using a higher number of points.


1	int	Points (must be a power of 2)
2	string	Output file name (or "")

Example. SPLINEFIT 1024 {c:\\Scan Data\\newfile.eeg}. The file in working memory will be spline fitted to 1024 points per second.

3.3.66 Statistical Comparisons

Statistical Comparisons.  Statistical comparisons between files/groups include correlations, t-tests, and z-scores.

3.3.66.1 CROSSCORR

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.

3.3.66.2 CROSSCORR_EX

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_EX "\$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.

3.3.66.3 INTRACLASSCORR

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:\\Scan Data\\Batch Examples\\reffile.avg" 100 200 "c:\\Scan Data\\Batch Examples\\corrfile.dat". The intra-class correlation is computed between 100-200ms, and the ASCII output file is created.


3.3.66.4 INTRACLASSCORR_EX

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").

1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	Boolean	Display results
5	string	Output file name (*.dat) (or "")
6	list	Channels to include

Example. INTRACLASSCORR_EX "c:\\Scan Data\\Batch Examples\\reffile.avg"
100 200 Y "c:\\Scan Data\\Batch Examples\\corrfile.dat" {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.

3.3.66.5 PAIRED

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:\\Scan Data\\Batch Examples\\output t-scores.avg". In the following example, we have already created the difference file (*diff file.avg*).

```
OPENFILE "c:\\Scan Data\\Batch Examples\\diff file.avg"
PAIRED "c:\\Scan Data\\Batch Examples\\output t-scores.avg"
```

3.3.66.6 PEARSONS

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:\\Scan Data\\Batch Examples\\referfile.avg" 0 500 "c:\\Scan Data\\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.

3.3.66.7 PEARSONS_EX

PEARSONS_EX - Pearson's r Correlation, extended. ~ The PEARSONS_EX command is the same as the PEARSONS command except for the new 5th and 6th parameters that let you show the results and 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_EX "c:\\Scan Data\\Batch Examples\\referfile.avg" 0 500 "c:\\Scan Data\\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.

3.3.66.8 TSCORE

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 (or "")
2	string	Output file name (or "")

Example. TSCORE "c:\\Scan Data\\Batch Examples\\compfile.avg" "c:\\Scan Data\\Batch Examples\\resultsfile.avg".

3.3.66.9 ZSCORE

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 (or "")
2	string	Output file name (or "")

Example. ZSCORE "c:\\Scan Data\\Batch Examples\\groupavg.avg" "c:\\Scan Data\\Batch Examples\\zscores.avg". The more complete script sequence might appear as:

```
set path "c:\\Scan Data\\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.

3.3.67 SUBTRACT


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:\\Scan Data\\Batch Examples\\compfile.avg" "c:\\Scan Data\\Batch Examples\\resultsfile.avg". Here is a simple example (be sure to use the OPENFILE command to retrieve the output file):

```
set path "c:\\Scan Data\\Batch Examples"
OPENFILE "$path\\sepnoblk.avg"
SUBTRACT "$path\\sepblk.avg" "$path\\difference.avg"
OPENFILE "$path\\difference.avg"
ZOOMIN "FZ"
```


3.3.68 SUBTRACTAVG

SUBTRACTAVG - Subtract AVG file from CNT file.  This command was taken from the EKG Noise Reduction transform. It is used to subtract an AVG file from each user selected event in the CNT file. It is usually used in conjunction with the CORRELATEPEAKS and QRS commands (see the EDIT manual for more detail).

1	string	Output file
2	defined value	Trigger type (RESPONSE, STIMULUS)
3	int	Trigger Code
4	string	Average file name
5	Boolean	Enable Dilate average
6	list	Channels to be modified (or "All")

Example. SUBTRACTAVG {D:\Scan Data\Demo Files\AudioVisual\EKG removed.cnt} STIMULUS 10 {D:\Scan Data\Demo Files\AudioVisual\EKG artifact.avg} Y {ALL}. The *EKG artifact.avg* file is subtracted from each stimulus type 10, from all channels, Dilate is enabled, and the new CNT file is saved.





3.3.69 TRUNCATECNT

TRUNCATECNT - Truncate CNT file.  This command is used to truncate a CNT file. The current CNT file will be overwritten to include the beginning of the file to the time point (in ms) that you enter. Be careful using this command - the original file will be overwritten! Make a copy of the original file first if you want to retain it.


1	double	Stop time (ms)
---	--------	----------------

Example. TRUNCATECNT 10000. The current CNT file will be overwritten to include the first 10 seconds only.

3.3.70 UPDATECHANGES

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).


3.3.71 VOLTAGETHRESHOLD

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 50 μ Vs (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).

3.3.72 Windowing Data Files

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_EX is the principle command.

3.3.72.1 EXCLUDEFORWINDOWDATA

EXCLUDEFORWINDOWDATA - Exclude Channel(s) for Windowing. Use this command to exclude channels from Windowing. This command is generally not needed if you are using the WINDOWDATA_EX command.

1...n list Electrode name(s)

Example. EXCLUDEFORWINDOWDATA {Fp1 Fp2 F7 F8}.

3.3.72.2 RESETFORWINDOWDATA

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. This command is generally not needed if you are using the WINDOWDATA_EX command.

Example. The RESETFORWINDOWDATA command is written on a line by itself, and uses no parameters.

3.3.72.3 WINDOWDATA

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:\Scan Data\windowed data.eeg". A Hanning window with a 10% taper is applied, and the new file is saved.

3.3.72.4 WINDOWDATA_EX


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").

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:\Scan Data\windowed data.eeg". All of the channels are selected for windowing.

3.3.73 WRITENUMSWEEPS

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:\\Scan Data\\Batch Examples\\numsweeps.dat".
The output file can be read with any text editor software.

4 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"
    # CLOSSES ALL OPEN FILES AND OPENS BASELINE CORRECTED EEG FILE FOR ARTIFACT
    # REJECTION
    # REJECTS EPOCHS WHERE PEAK AMPLITUDE EXCEEDS +/- 150 microVolts between 0 and
    400ms
    # 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"

    # CLOSSES 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:\\Scan Data\\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
    ARTREJCRITERIA Y X X Y $nmv $pmv Y N
    set numrej 0
    for {set epochnum 0} {$epochnum < $evcount} {incr epochnum} {

```

```

        if {[GETEPOCHINFO $epochnum -ACCEPT] == 0} {
            incr numrej
        }
    }
    set PercentRej [expr {100.0*$numrej/$evcount}]
    set extname $mv
}
INSTRUCT "percentage reject: $PercentRej"
SAVEAS "c:\\Scan Data\\Batch Examples\\$name$extname.eeg"
OPENFILE "c:\\Scan Data\\Batch Examples\\$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:\\Scan Data\\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\Acquire5\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
```

5 Quick Reference Guide





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 or COH files.

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.

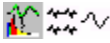
- | | | |
|---|------|----------------------------|
| 1 | list | Electrode name(s) or "ALL" |
|---|------|----------------------------|

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. You must first retrieve one file first. The files listed in the command will be appended to it, creating a new file.


1	list	File(s) to append
2	string	Output file name

AREAREPORT⁶² - 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.


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)

ARRANGEWINDOWS⁶⁷ - 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)
---	---------------	--------------------------------------


ARTCOR⁶³ - 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. If you are using an EEG file instead of a CNT file, use the 8th parameter for the corrected output file path and name.

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)



ARTREJ⁶³ - 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 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_EX⁶⁴ - 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

ARTREJCNT⁶⁴ - 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 has a somewhat different use in this case (see the EDIT manual for 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 (μV)
6	double	Max amplitude (μV)
7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skip channels
9	list	List of rejection electrodes

ARTREJCNTFREQ⁶⁵ - 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 (μV)
6	Boolean	Recompute
7	Boolean	Exclude Bad channels
8	Boolean	Exclude Skipped channels
9	list	List of electrodes to include


ARTREJCNTTRIG⁶⁶ - Artifact Rejection for CNT files (using Triggers).   Regions around trigger events in CNT files can be rejected. A pre-trigger interval would be, for example, pre: -500 and post: -400. A post-trigger interval would be, for example, pre: 100 and post: 200. A symmetric interval about the trigger would be, for example, pre: -100 and post: 100.

1	double	Pre-trigger (ms)
2	double	Post-trigger (ms)
3	int	Trigger code
4	Boolean	Recompute

ARTREJFREQ⁶⁶ - 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 number 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 - 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. 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


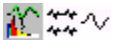
AVERAGE_EX - Average Sweeps, Extended. This command is the same as the AVERAGE command except for the way in which the SNR is computed. You have control over the options used for the intervals (see the EDIT manual for more details).

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


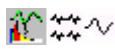
AVGBANDS - Modify Frequency Bands in AVG Files. (frequency domain AVG and COH files only). This option formerly existed as the AVGBAND.exe utility program for the DOS version of

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.


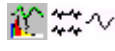
1	string	Output band file name
x	string	Band label
x	double	Band start frequency
x	double	Band stop frequency

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  - 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"). The "Auto-write" feature will create the 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)

BASECOR_EX2  - Exclude Bad or Skipped Channel(s) for Baseline Correction.  This command differs from the BASECOR_EX command in that it allows you to exclude Bad or Skipped channels. The "Auto-write" feature will create the command automatically using the Exclude options you specify.



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	Boolean	Exclude Bad channels
5	Boolean	Exclude Skipped channels
6	list	List of electrodes to correct, or "All"
7	string	Output file (not needed for AVG files)



BLINKNOISEREDUCTION  - Blink Noise Reduction Routine.  The BLINKNOISEREDUCTION command is intended for the removal of VEOG blink artifact from CNT files. 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 (μ Vs; 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 (μ V)
12	double	Artifact max (μ V)
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

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


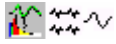
1	Boolean	Enable butterfly plot
---	---------	-----------------------



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  ⁷⁴ - 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  ⁷⁵ - 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).




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  ⁷⁵ - 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 "")

COMPAREELECTRODES  ⁷⁶ - Compare Electrodes.   This command allows you to compare up to 8 electrodes in a Multiple Window display. (See also REMOVECOMPAREELECTRODES).

1	string	Primary Electrode name
2	string	List of Compare Electrode name(s) or All.

CONCATCNT  ⁶¹ - 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 "")

CORRELATESPEAKS  ⁷⁶ - Correlate Peaks.   The CORRELATESPEAKS command was taken from the EKG Noise Reduction transform. It performs a series of intra-class correlations in order to align peak events by waveform similarity rather than event code placement (see the EDIT and MagLink RT manual for more details).


1	defined value	Trigger type (STIMULUS, RESPONSE)
2	int	Type code
3	double	Correlation interval Start (ms)
4	double	Correlation interval End (ms)
5	double	Shift Limit (points)
6	int	Averages
7	string	Correlation Channel label
8	defined value	Bad peaks (PROCESS, REMOVE, CHANGECODE)
9	double	New event code (ignored if 8 is PROCESS or REMOVE)
10	double	Correlation Threshold (ignored if 8 is PROCESS)

CREATESORT⁷⁷ - Creates a Sort.  See the CREATESORT command (in the body of this manual) for complete details, as well as DELETESORT.


1	string	Sort name
---	--------	-----------

CROSSCORR¹²⁴ - 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_EX¹²⁵ - 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

CUTEPOCH⁷⁸ - 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

DCCORRECT⁷⁸ - DC Correction.  The DC Correction option will correct 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

DEBLOCK  - Apply Deblocking.  The DEBLOCK option is used to replace stimulus artifact with a flat line. Windowing is performed to reduce any abrupt transitions at the beginning and end of the lines.



1	defined value	Window Type (NO_TAPER, COSINE, BLACKMAN, HANNING, HAMMING, PARZEN, WELCH)
2	double	Taper percentage
3	int	Trigger code (CNT files only)
4	double	Start latency (EEG and AVG time domain files only)
5	double	End latency (EEG and AVG time domain files only)
6	list	Electrode name(s) to correct, or "All"
7	string	Output file (CNT and EEG files only)

DECIMATE  - Decimate and Apply Noise Suppression.  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 can be applied (passed four times) to correct for aliasing.



1	int	New AD Rate
2	Boolean	Enable Filter
3	float	Low pass value
4	int	Slope (poles; 0=6dB/oct, 1=12dB/oct, 2=24dB/oct, 3=48dB/oct)
5	string	Output file

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 CREATSORT.




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  ⁸¹ - Linear Detrend.   This option is used to remove linear drifting components.




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  ⁸¹ - 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"). The "Auto-write" feature will create the 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  ⁵³ - 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  ⁵³ - Do Impedance Testing.  This command will initiate impedance testing.

EKGNOISEREDUCTION  ⁶⁷ - EKG Noise Reduction.   The EKGNOISEREDUCTION command is used to reduce EKG and BCG artifact in CNT files. The original command was also EKGNOISEREDUCTION. This was modified with the EKGNOISEREDUCTION_EX command, and then again with the EKGNOISEREDUCTION_EX2 command. For the V4.4 release, we are returning to the original EKGNOISEREDUCTION command. This will include all of the settings used in the V4.4 version of the software. If you have an existing batch file that uses a prior version of the EKGNOISEREDUCTION command (or _EX, or _EX2), you will need to replace it with the V4.4 version of the EKGNOISEREDUCTION command. EKGNOISEREDUCTION_EX and EKGNOISEREDUCTION_EX2 are obsolete and will not be recognized. If you have an existing batch file with the original EKGNOISEREDUCTION command, you will get an error message when you try to run it with V4.4 (incorrect number of parameters). In other words, *you will need to replace any prior batch command for EKG Noise Reduction with the new version of the EKGNOISEREDUCTION command.*

Because of the large number of parameters used with this transform, it is recommended that you set the parameters in point & click mode, apply the transform, then copy and paste the line into the batch file you are creating.


1	string	Output file
2	double	Artifact Epoch Start
3	double	Artifact Epoch Stop
4	integer	Number of averages
5	defined value	Trigger type (QRS, VOLTAGE, RESPONSE, STIMULUS)
6	int	External Trigger Code
7	double	Voltage Threshold (μ V) (ignored if not using voltage triggering)
8	defined value	Trigger direction (POSITIVE, NEGATIVE)
9	string	Trigger channel label
10	Boolean	Insert Events
11	int	Insert Code
12	Boolean	Enable Trigger channel Filter
13	defined value	Filter Type (LOWPASS, HIPASS, BANDPASS, BANDSTOP)
14	float	High Pass cutoff
15	int	High Pass attenuation (db)
16	float	Low Pass cutoff
17	int	Low Pass attenuation (db)
18	float	Notch start frequency
19	float	Notch stop frequency
20	int	Notch attenuation (db)
21	double	Refractory period
22	Boolean	Enable Correlation
23	double	Correlation Interval Start
24	double	Correlation Interval Stop
25	int	Shift limit (data points)
26	double	Correlation threshold
27	string	Correlation channel label
28	Boolean	Enable Dilate average

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 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

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

ERBP  ⁸³ - 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 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

ERCOH  ⁸⁴ - 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^[73] - Exclude Channel(s) for Baseline Correction.  Use this command to exclude channels from Baseline Correction. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.

1 list Electrode name(s)


EXCLUDEFORDETREND^[81] - Exclude Channel(s) for Linear Detrend.  Use this command to exclude channels from Detrending. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.

1 list Electrode name(s)


EXCLUDEFORFILTER^[95] - Exclude Channel(s) for Filtering.  Use this command to exclude channels from Filtering. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.

1 list Electrode name(s)


EXCLUDEFORPEAKDETECTION^[110] - Exclude Channel(s) for Peak Detection.

 Use this command to exclude channels from Peak Detection. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.


1 list Electrode name(s)

EXCLUDEFORSMOOTH^[118] - Exclude Channel(s) for Smoothing.  Use this command to exclude channels from Smoothing. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.



1 list Electrode name(s)

EXCLUDEFORWINDOWDATA^[130] - Exclude Channel(s) for Windowing.  Use this command to exclude channels from Windowing. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.



1 list Electrode name(s)

EXPORTAVG^[85] - 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  - 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)

EXPORTAVG_EX2  - Export AVG files, Extended.  Comma Delimiting has been added to the EXPORTAVG_EX command. Comma delimiting separates the data points by commas, rather than by tabs. The parameters for EXPORTAVG_EX2 are:

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	Boolean	Use comma delimiting
14	list	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  - 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)

EXPORTCNT_EX2  - Export CNT File to ASCII, Extended.   Comma Delimiting has been added to the EXPORTCNT_EX command. Comma delimiting separates the data points by commas, rather than by tabs.

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	Boolean	Use comma delimiting
12	list	Channels to be exported (or "all")

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 (created in the Montage Editor) 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

EXPORTCOH_EX2  - Export COH files, Extended.   Comma Delimiting has been added to the EXPORTCOH_EX command. Comma delimiting separates the data points by commas, rather than by tabs.

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
6	Boolean	Use comma delimiting

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
4	list	Channels to be exported

EXPORTEDF_EX⁹⁰ - Export 32 bit data to EDF.   The EXPORTEDF command was modified to provide various scaling methods that can be used when exporting 32 bit data files to the 16 bit EDF format (see the EDIT manual for details).



1	string	Output file
2	Boolean	Include Bad channels
3	Boolean	Include Skipped channels
4	list	Channels to be exported
5	defined value	Scaling method (AMPLIFIERRESOLUTION, PERCHANASYMMETRICAL, PERCHANASYMMETRICAL, ALLCHANASYMMETRICAL, ALLCHANASYMMETRICAL, USER)
6	double	User scaling Min
7	double	User scaling Max
8	defined value	Block size (FIXED, AUTO)
9	double	Block seconds

EXPORTEEG⁹⁰ - 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_EX⁹⁰ - 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)

EXPORTEEG_EX2⁹¹ - Export EEG File to ASCII, Extended.   Comma Delimiting has been added to the EXPORTCOH_EX command. Comma delimiting separates the data points by commas, rather than by tabs.

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	Boolean	Use comma delimiting
14	list	Channels 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

EXTRACTBLOCK  - Extract Block from CNT file.  This command is used to create new files from blocks of data extracted from CNT files. Blocks can be defined by stimulus or response events, or time points. Multiple output files may be created. Each will have a suffix added automatically (*_1.cnt, *_2.cnt, *_3.cnt etc.).

1	defined value	Method (STIMCODE, RESPONSECODE, INTERVAL, TIME)
2	integer	Event Code (STIMCODE and RESPONSECODE only)
3	double	Interval (ms) (INTERVAL only)
4	double	Start time (ms) (TIME only)
5	double	End time (ms) (TIME only)
6	string	Output file (suffixes may be appended)

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(opt)	string	Output file name, or "" (omitted for AVG files)

FILTER_EX  - 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"). The "Auto-write" feature will create the 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)

FMRI - fMRI Artifact Reduction. The fMRI transform averages the MR slice artifacts

(creating an average for each channel), and then subtracts the average from those sections containing the slices. The average start and stop points are determined by the placement of trigger events (see the EDIT manual for more details). The transform is included with the Maglink RT system (contact techsup@neuroscan.com for details).

1	int	TR Trigger code
2	string	Trigger Method (VOLTAGE, STIMULUS, RESPONSE)
3	double	TR start (ms)
4	double	TR duration (ms)
5	int	Number of slices per block
6	int	Number of Averages
7	defined value	Decimation Frequency (Hz)
8	Boolean	Enable Correlation
9	int	Shift limit (in sample points)
10	Boolean	Enable Continuous Removal
11	Boolean	Enable Filter
12	float	Low pass filter cutoff (Hz)
13	int	Offset (in points) to Voltage Threshold
14	float	Voltage Threshold
15	int	Preview channel number
16	string	Output file

FORWARDDFFT - Forward FFT. The FORWARDDFFT 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 CREATSORT.



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 - 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"). The "Auto-write" feature will create the 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.

1(opt) string AST file name (or "")



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.






GETDATE  / **GETTIME**  -  Get the current date and time. Use these commands

to return the current date and time (using the computer clock).




Return. The commands will return the current date and time.

1 Boolean Use formatting




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.

GETINPUTFILE³⁷ - Get Input File.  This command is used to display the standard Open File utility display, with optional customization. See also GETOUTPUTFILE.

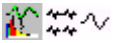
1	string	String title
2(optional)	string	Extension
3(optional)	string	Initial Path

Return. The full path of the file selected by the user.

GETMARKERCOUNT¹⁰² - 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.


GETMARKERINFO¹⁰³ - 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.

GETNUMCHANS³⁸ - 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.

GETNUMDISPLAYPAGES³⁸ - 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.


GETNUMPOINTS³⁹ - 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.

GETNUMSWEEPS⁴⁰ - 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)

GETOUTPUTFILE⁴⁰ - Get Output File.  This command 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.


GETPOINTDATA¹⁰⁴ - 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 only)	int	Epoch number (zero-based index)

GETSWDISPLAYSPEED⁴¹ - 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.

GETSWEEPMAX⁴² - 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)

GETSWEEPMIN⁴² - 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)


GFP¹⁰⁵ - 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

GROUPAVG⁷¹ - 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

HERTZTOPOINT⁴²

- 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.

HIGHIMPEDANCE⁵⁴

- 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

IMPORTAVG⁹²

- 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

IMPORTCNT⁹²



- 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 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
2	int	Number of channels
3	double	Acquisition rate
4	defined value	Company (NEUROSCAN, NK)



IMPORTEEG⁹²

- 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



IMPORTEEG_EX  - Import EEG File from ASCII.  In the prior IMPORTEEG batch command, X Minimum was specified in *seconds*. A new batch command has been added: IMPORTEEG_EX. The parameters are the same as before, only with this command X Minimum is in *milliseconds*.

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



IMPORTEVENT  - Import Event File.  This command is used to import an event file (.evt or .ev2) into a CNT file. The event file can replace or be added to the existing event table, and the events can be in byte offsets or second offsets.

1	Boolean	Replace events
2	Boolean	Use seconds
3	string	Event file name



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.

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

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)

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  ¹⁰⁸ - Integrate Waveform.  This command performs an accumulating sum across each sweep for each channel (refer to the EDIT manual for more details).


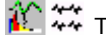
1	Boolean	Rectify
2	string	Output file (required for AVG and EEG files)

INTRACLASSCORR  ¹²⁵ - 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  ¹²⁵ - 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"). The "Auto-write" feature will create the command automatically using the electrodes you specify in the "Select" dialog box (or "All").


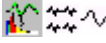
1	string	Reference file name (or "")
2	double	Correlation interval start
3	double	Correlation interval end
4	Boolean	Display results
5	string	Output file name (*.dat) (or "")
6	list	Channels to include

INVERSEFFT  ¹²² - 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
---	--------	------------------

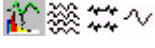
ISOFFLINE  ⁴⁴ / **ISONLINE**  ⁴⁴ - Return Online or Offline.  These commands are used primarily in conjunction with the autorun feature. You may want to have a different set of functions loaded depending on whether you are online (in ACQUIRE) or offline (in EDIT). The commands will return either a 0 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.

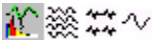
LATENCYTOPOINT  ⁴⁴ - Latency to Point Index.  This command is used to return the index number of the data point that corresponds to a given latency (AVG and EEG files only). For example, the GETPOINTDATA command is used to return the value of a given data point, or set of points. It requires the index 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)

LDR¹⁰⁸ - 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)

LOAD3DD¹⁰⁹ - 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 "")

MAXIMIZE⁴⁵ - 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.

MINIMIZE⁴⁵ - 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.

MERGE⁶¹ - Merge.  This command is used to append one or more EEG files to a previously opened file. The files listed in the command will be merged to it, creating a new file. The command is applied only to epoched (EEG) files.

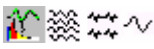
1	list	Files to merge
2	string	Output file name

MERGEevt¹⁰⁹ - 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.


1 string Task file name

NEWNAME³¹ - 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)

OPENFILE¹⁰⁹ - 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

PAIRED¹²⁶ - 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.

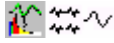
1 string Output file name

PAUSE⁴⁵ - 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.

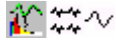
1 (optional) int Time to wait (in ms's)

PAUSEDISPLAY⁵⁴ - Pause Display.  This command is used to pause or un-pause the display without interrupting data storage.

1 Boolean Pause display

PEAKDETECTION¹¹¹ - Peak Detection.  This command performs an automatic peak detection placing markers at those locations.


1	string	Marker label
2	Boolean	User defined interval
3	float	Start search point (ignored if param 2 is FALSE)
4	float	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_EX¹¹¹ - 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"). The "Auto-write" feature will create the 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")

PEARSONS¹²⁶ - 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_EX¹²⁷ - 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.


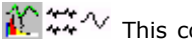
PICKOFFNAMEONLY c:\scan4.3\Batch Examples\viscpt.cnt yields viscpt.

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

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	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.

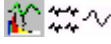
PRINTWAVEBOARD  - Print Waveboard Display.  This command is used to Print the display in the Waveboard (no additional parameters are used).

QRS¹¹³ - QRS Detection.  This command will perform an automated detection of QRS complexes and insert a stimulus type code for each.


1	string	Trigger Channel	1	string	Trigger Channel
2	int	Stim code	2	int	Stim code

READPOS⁴⁸ - 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.

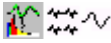
1	string	Position file name (ASC extension)
---	--------	------------------------------------

READREJ⁸⁹ - 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
---	--------	---------------------------

READSUB⁵⁰ - 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.

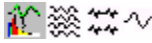
1	string	Subject file name
---	--------	-------------------

RECTIFY¹¹⁴ - 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_EX¹¹⁴ - 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(opt)	string	Output file name (or "")




REFER¹¹⁴ - 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




REJECTBLOCK¹¹⁵ - Reject Block.  Use this command to reject blocks of data in CNT files. (You will need the AD rate to compute the offset: offset = AD/1000 x 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.

REMOVECOMPAREELECTRODES  - Remove Compared Electrodes.   This command removes all or selected electrodes that have been compared with the COMPAREELECTRODES command.




1	string	Primary Electrode name
2	string	Compare Electrode name(s) or All to be removed

REMOVEEVENT  - Remove Event from CNT File.   This command is used to remove events from CNT files, one event at a time.




1	int	Index of event (zero-based)
---	-----	-----------------------------




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. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.


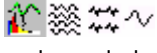
RESETFORDETREND  - Reset All Channels for Linear Detrend.   This command will reset all channels for linear detrending (so that none are excluded). It is typically used where some channels had been excluded for detrending. It avoids having to reset each channel individually. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.



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. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.

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. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.



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. This command is generally not needed if you have a newer version of software that uses the _EX or _EX2 command, in which the channels are listed in the command itself.

RESETFORWINDOWDATA  ¹³⁰ - 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. This command is generally not needed if you have a newer version of software that uses the `_EX` or `_EX2` command, in which the channels are listed in the command itself.


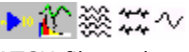
RESPWIN  ¹¹⁶ - Response Window.  The RespWin (Response Window) program performs an offline analysis of behavioral data in an event (.EVT) file that consists of a stream of stimulus and response events. The user defines a response window and one or more paired subsets of stimuli that correspond to target 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 parameters 7-11 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
7	string	Descriptor
8	string	Target vector (1, 2, 3-6, etc.)
9	int	Target correct response
10	string	Distractor vector (1, 2, 3-6, etc.)
11	int	Distractor correct response (or 0)


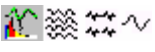
RESTART  ⁵⁵ - Restart Acquisition.  This command serves the same purpose as the Restart icon on the ACQUIRE Toolbar, is used to restart the acquisition process.

RESTORE  ⁴⁹ - Restore Window.  This command performs the standard Windows Restore function. Applying it will Restore the display of the window that has the focus. Use SELECTFILE to change the focus to the desired window, as needed.


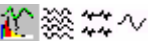
Example. RESTORE. No additional parameters are used.

REVIEW  ⁴⁹ - 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.


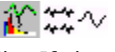
1 (opt) int Time to wait (in milliseconds)

SAVEAS  ¹¹⁰ - 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


SAVEAS_EX  ¹¹⁰ - Save File As, with DC option.  This is the same as the SAVEAS command, with the additional option to save DC offset and drift corrections permanently (following application of the DC Offset Correction transform; CNT files only).

1	string	Output file name (or "")
2	Boolean	Make DC correction permanent


SAVEPEAK  ¹¹² - 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 (opt) | defined value | option (-SubstituteBad) |
| 3 (opt) | string | Substitution string |

SAVEPOS⁴⁸ - 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) |
|---|--------|------------------------------------|

SAVEREJ⁷⁰ - 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 |
|---|--------|--------------------------|

SAVESUB⁵⁰ - 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) |
|---|--------|---------------------------|

SAVEEVENT¹¹⁷ - 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_EX¹¹⁷ - 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 "") |

SCALE³⁰ - 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^[30] - 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^[118] - 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^[31] - 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^[33] - 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^[76] - Set the Coherence Reference.   (COH files only) This command is used to set or change the Coherence Reference.

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

SETCURRENTEPOCHINDEX^[97] - 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^[35] - Set Display Inversion.    This command is used to set the display polarity. See also GETDISPLAYINVERTED.



1	Boolean	Inverted
---	---------	----------

SETDISPLAYPAGE^[36] - 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^[98] - 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^[100] - 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^[50] - 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^[41] - 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^[119] - 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(opt)	string	EEG output file name (omitted for AVG files)

SMOOTH_EX^[119] - 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"). The "Auto-write" feature will create the 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

SMOOTH_EX2^[120] - Display Dialog for Smooth.   This command differs from the SMOOTH_EX command in that it allows you to display the Smoothing dialog window before the transform is applied. 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	Boolean	Show dialog
5(opt)	string	EEG output file name

SNR  ¹²⁰ - 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  ¹²⁰ - 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  ¹²¹ - 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
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
10	Boolean	Derive channels for unwanted signals
11	Boolean	Derive channels for Signals of interest
12	string	Output file name (CNT files only)

SPATIALSVD  ¹²² - 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

SUBTRACTAVG¹²⁸ - Subtract AVG file from CNT file.  This command was taken from the EKG Noise Reduction transform. It is used to subtract an AVG file from each user selected event in the CNT file. It is usually used in conjunction with the CORRELATEPEAKS and QRS commands (see the EDIT manual for more detail).


1	string	Output file
2	defined value	Trigger type (RESPONSE, STIMULUS)
3	int	Trigger Code
4	string	Average file name
5	Boolean	Enable Dilate average
6	list	Channels to be modified (or "All")

TOGGLEZOOM⁵² - 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
---	--------	----------------


TRUNCATECNT¹²⁹ - Truncate CNT file.  This command is used to truncate a CNT file. The current CNT file will be overwritten to include the beginning of the file to the time point (in ms) that you enter. Be careful using this command - the original file will be overwritten! Make a copy of the original file first if you want to retain it.


1	double	Stop time (ms)
---	--------	----------------

TSCORE¹²⁷ - 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

UPDATECHANGES¹²⁹ - 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.


VIEW⁵⁸ - 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.

VOLTAGETHRESHOLD¹²⁹ - 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

WINDOWDATA¹³⁰ - 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(opt)	string	Output file name

WINDOWDATA_EX¹³⁰ - 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"). The "Auto-write" feature will create the 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 "")

WRITELOG⁵¹ - 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

WRITENUMSWEEPS¹³¹ - 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
---	--------	-----------

ZOOMIN⁵² - 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
---	--------	----------------

ZOOMOUT⁵² - 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
---	--------	----------------

ZSCORE¹²⁷ - 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