

1 New to ERTS V3.36

1.1 Multiple Session definitions

You can now define different session structures for the same experiment by creating multiple session definitions within the same ERTS script.

Syntax: `SESSION [Name]`

`Name` Name of Session which is used also as short description within the results file

In order to be compatible to previous ERTS releases, the name is optional but must be defined if multiple sessions are defined. Only the token up to the next delimiter (blank, comma, ..) will be used as the name.

If multiple sessions are defined, you must either assign session definitions to subject groups (see new `GROUP` command) or ERTS will randomly select one of the session definitions when you launch an experimental session. The name of the selected session is displayed before starting the experimental session and the session number is included in the result file.

1.2 Subject groups

In order to automate between-subject designs, the new `GROUP` command supports the definition of subject groups.

Syntax: `GROUP Name`
`SessName SubjNo1 [SubjNo2 [.. SubjNoN]]`

`Name` Name of the group. This will be listed in the result file
`SessName` Name of the session definition which should be launched for this group
`SubjNo` Subject numbers that belong to this group

A group is defined by simply enumerating the subject numbers that go into this group. The group is then assigned to a session definition and depending on the subject number, the corresponding session definition is selected. ERTS will display the selected group after you have typed in the subject number

Note that the `GROUP` command must be placed behind the session definitions because the command is referring to these definitions.

1.3 Choosing blank character for screen echo

In the `SE` trial command, you can now specify any symbol in addition to the underscore as the blank character

Syntax: `SE Position [PosNo [Font [Case [FieldChar]]]]`

`FieldChar` 0 for blank
1 for underscore
> 1 decimal value of the ASCII character to be used as fieldchar

1.4 Limiting key coding to registration period

The CODEKEYS (only ERP-Version) introduces an additional parameter for limiting key coding (sending sync signals upon key presses) to periods of active response registration (RK-command).

Syntax: CODEKEY [Channel Keys ByteSequ [ToneSet [RegOnly]]]

RegOnly 0 code any key action during session
 1 code only key action within active response registration periods

1.5 Managing bonus points

For counting and displaying bonus points, ERTS has a special internal numerical bonus counter. This counter can be initialized or increased/decreased using the BO-trial command

Syntax: BO Operation Amount [FbWidth Precision]

Operation 0 Initialize internal counter with amount
 1 Add amount to current value
 2 Subtract amount from current value
 3 Add Rt dependent value (value = Rt / value)

Amount Numerical positive value. If precision is larger than 0 then the n right digits will be interpreted as decimal places (e.g. if precision=2, then 1050 will be interpreted as 10,50)

FBWidth 0: Do not store bonus value in FB column
 n > 0: Store bonus value in FB column and set width of FB column to n characters.

Precision 0: Interpret value as defined
 n > 0: Interpret the n right numbers as decimal places

The default width of FB columns is 3 characters. If your bonus points are greater or equal to 1000 than you need to set width of 4 or higher.

1.6 Displaying bonus points

Bonus points are displayed using special placeholders in a formatted picture definition (see PICTURE command with subtype F)

@BBBB Display last amount
 @bbbb Display current points

1.7 Displaying time of block

Additional place holders in formatted pictures support the display of the elapsed or residual block execution time.

@tttt Display elapsed block execution time in seconds

@TTTT Display remaining block execution time in seconds (MAXTIME must be active)

1.8 User defined token delimiter for single word presentation

The following session command defines a token delimiter other than blank

```
SETSWDELIM      AsciiVal

      AsciiVal  Numerical Value which is the decimal equivalent of the character that should be used as
                  the delimiter
```

1.9 Extended CS-command for equi-distant random time interval

In addition to the exponential random time interval, the CS-command now supports the definition of an equi-distant random time interval. Whenever MaxRnd is less or equal to RndTime, ERTS will interpret these values as follows:

```
RndTime  Time range that should be varied at random

MaxRnd   Divisor for creating equi-distant time intervals within this range
```

For example, if MaxRnd=4 and RndTime=4000 then five different time values are generated at random: 0,1000,2000,3000,4000.

1.10 Defining Sets of Sets

The SET command now supports the definition of sets that exists of sets. The new sub-type for this new type of element is 'E'.

1.11 Extended RK-command

The RK-command has been extended by two arguments

Syntax: RK [Key [MaxRt [N(Keys) [Skey [EndKey [Echo [Reset AutoRep [DblKey]]]]]]]]

```
AutoRep  Auto repetition mode for PC-keyboard and EXKEY. For PC-keyboard the autorep rate
          is determined by BIOS setting. For external keys (EXKEY), the value sets the start
          delay in msec. If a key is pressed longer than AutoRep msec, ERTS will start to auto-
          repeat that key press with increasing rate. (default is 0, no auto rep).

DblKey   If you register multiple responses (N(Keys) > 1) and this flag is set to zero, each key
          can only be used once. (default 1)
```

1.12 Removing parallel element

If no element is specified in the RE-command, then the element with the same index that has been used by the most recent DE-command will be removed from the set. This is similar to parallel drawing of elements, but instead the parallel element will just be removed.

1.13 Start message and warning tone for unstable tracking

The RUNUNSTABLE command has been extended to support the display of a start message at the beginning of the tracking task and the presentation of a warning tone prior to hitting the borders.

Syntax: RUNUNSTABLE T(WarmUp) T(Record) [Lambda [Gain [Range [Key [Compress [Tone] [StartPic] [StartPos] [StartFont] [WarnRange]]]]]]]]]]

StartPic	The name of the picture definition that should be displayed as the starting message at the beginning of the tracking task
StartPos	The name of the position definition on which the start message should be displayed
StartFont	The name of a font definition that should be used to render the message
WarnRange	The “quiet” tracking range [mm/10]. If this range is exceeded by the vertical bar a warning tone will be presented as defined in the ‘Tone’ argument.

1.14 Selective feedback for dual task situations

The OK-command has been extended to support selected evaluations of responses with respect to time and key order.

Syntax: OK CompMode KeyIndex LowTime HighTime LowKey HighKey

Low/High Time	Limits the evaluation to responses that are greater than LowTime and lower than HighTime. All other responses will be ignored
Low/High Key	Limits the evaluation of responses to key presses that are inbetween LowKey and HighKey. All other responses will be ignored.

When cutting the evaluation range by defining a LowKey ERTS interpret all following latency relative to LowKey. This feature allows you to look at latencies relative to another keystroke. If both a key and a time range is defined, ERTS will first determine the key range and then will determine the time range relative to LowKey.

Typically, LowKey will refer to a key that is generated by the GR trial command to time stamp the onset of a stimulus within the response interval. This way the OK command will only interpret responses that are registered after the onset of that stimulus and will interpret latencies relative to the onset

Unlike FB you can use OK multiple times with different parameters. This enables you to evaluate all four error conditions one after the other. With +OK you could even display all messages at once if you use different screen positions.

1.15 Immediate start of dual task

The RUNMSEARCH and SETMSEARCH command has been extended by the AutoStart argument to support the immediate start of the task without displaying a target set and without waiting for further confirmation by the subject. This is useful if the task must be synchronized with EEG/MRI devices and when this command is used to display plain lists instead of a memory task (see ListMode argument).

Syntax: RUN/SETMSEARCH N(Valid) N(WarmUp) TargSet ProbeSet TargPos ProbePos StartKey ò YesNoKey MaxRt Pause [Sequence [Port [ParSet [ForceEndOfTracking [ListMode] [AutoStart]]]]]

AutoStart	0: Task is started by key press (default)
	1: Task is started automatically

1.16 Periodic syncing with external counter

The WC trial command (Wait for external Counter) has been extended to support a periodic syncing on every i-th trial

Syntax: WC [[NTics [ResetCounter [Cycle]]]

Cycle Integer value that defines the number of repetitions in which the WC command should be executed. For example, a value of '2' means that every second time the WC command will wait for an external counter value.

Use this new argument if you want to run several trials in-between two external MRI-signals but still want to re-calibrate with each new signal.

1.17 Writing all results at end of session

By default, ERTS is storing results after each block. If your block transitions are time critical, use the new /J command line option to hold all results in memory until the end of the session. This will, of course, require more working memory than the block-wise writing of results.

This new option does not work in combination with tracking tasks and staircases.

1.18 Analog joystick response in ERTS

ERTS provides some support for analog joystick responses that are based on movement. The registration of the analog joystick must follow different principles because the game port does not produce interrupts. ERTS must poll the game port in order to detect any changes.

Because of this, the time resolution depends on CPU speed and the response characteristic of the game port. You can test this speed by checking the port with the TESTJOY utility. In average, response time should be within a 5ms range.

Unlike with the RK command there exist some same restrictions for MM and MJ. Only the next trial command is executed and then ERTS waits for a joystick response. There is no continuation and no internal pre-loading of images before the response is made or MaxRT is expired.

There are two ways to think about joystick movements. One is in terms of the angle the joystick is moved away from the center position. The second is in terms of screen coordinates where the joystick cursor is located.

For recording joystick movements as a response, the following trial commands have been implemented:

MM Record the leaving of a center position and crossing of circular boundaries
MJ Record the leaving of a starting zone and arrival on target zone

Please note that all features described in this chapter are provided on an "as is" base

1.19 MM (Measure Movement)

The MM command measures the time until the joystick is moved across user-defined boundaries. Key names are assigned to each boundaries in order to code the event in the result file.

Syntax: MM Radius Angle Tolerance Keys [MaxRt [Cursor [ResultMode [Scale]]]]

Radius	Numeric set definition enumerating the boundaries which trigger a response when crossed (from center to outside). A negative value means that any crossing triggers a response. A positive value triggers a response only if the crossing is within a valid sector (see angle and tolerance)
Angle	Numeric set definition enumerating the angles that define valid sectors
Tolerance	Tolerance defines the size of the valid sectors angle plus/minus tolerance).
Keys	A KEY definition that is used as a translation table for the angles. List of key names within KEY definition and list of angle in numeric key set must correspond.
MaxRt	Maximum response time
Cursor	Name of a SET definition consisting of at least on picture which will be used as the joystick cursor Valid picture types are P, G, R.
ResultFormat	Determines the format of the result file <ul style="list-style-type: none"> 0 X/Y-Coordinates (default) 1 Radius and Angle
Scale	Determine metric of results for radius <ul style="list-style-type: none"> 0 Screen coordinates 1 Joystick angle (see STICKANGLE)

1.20 MJ (Measure Joystick)

This command measure the time of leaving a starting zone and the arrival of one or more target zones. The target zones are mapped to key names.

Syntax: MJ StrtRadius TargPos PosNo PosRadius Keys [MaxRt [Cursor [Scale]]]

StrtRadius	Radius which triggers first response
TargPos	Position definitions that define valid target zones
PosNo	Index of position (within the position definition) which is used as target zone. If zero, all positions of TargPos are valid.
PosRadius	Defines the size of the valid area around each target zone
Key	KEY definition that is used as mapping table for leaving the start zone and target zones. First key is used for leaving the target zone. The remaining keys are mapped to the target locations in ascending order.
MaxRt	Maximum response time (like RK)
Cursor	SET Definition that contains at least two pictures. The first picture is used outside of valid zones, the second on target zones. Valid picture types are P, G, R.
Scale	Determines metric of radius <ul style="list-style-type: none"> 0 Screen position 1 Angle of joystick position (see STICKANGLE)

1.21 WJ (Wait for Joystick Position)

This command waits until the joystick is positioned on a target position for a given amount of time. Note that this event is not recorded.

Syntax: WJ TargPos PosNo [PosRadius [OnTarget [Cursor [Scale]]]]

TargPos	POSITION definition that defines the target position
PosNo	Index of position within TargPos that is used as target position
PosRadius	Radius around TargPos that defines the size of the target area
OnTarget	Time in milliseconds which defined the duration at which the cursor must be on target before the response is accepted
Cursor	Set definition that contains at least two pictures The first picture is used outside of valid zones, the second on target zones. Valid picture types are P, G, R.
Scale	Determines metric of radius <ul style="list-style-type: none"> 0 Screen position 1 Angle of joystick position (see STICKANGLE)

1.22 Additional columns in result file

Two new columns 'Wn' and 'Rn' will be generated when you use the MM or MJ commands. For each radius one column will be produced

Rn	Latency of crossing boundaries
Xn	X-coordinate of joystick
Yn	Y-coordinate of joysticks
Dn	Radius of joysticks
Wn	Angle of joystick
Tn	Key name (mapped to target zones)

1.23 Using Joystick keys

The WK command also handles J1 and J2.

The RK command also handles J1 and J2 to measure latencies but the same restrictions exist like for MM and MJ which means that only the next trial command is executed and then ERTS waits for a joystick response.

There is no continuation and no internal pre-loading of images before the response is made.

1.24 Calibration of joystick coordinates

The SETSTICK session command and the AS trial command have been extended to support vertical calibration as well

Syntax: SETSTICK center left right top bottom

Syntax: AS Key PosFlag

PosFlag	Determines what position will be checked
1	left (Xmin)
2	center (X/Y middle)
3	right (Xmax)
4	top (Ymin)
5	bottom (Ymax)

The new session command STICKANGLE can be used to set range of overall movement in terms of angle position

Syntax: STICKANGLE Xangle [Yangle]

Xangle	Overall horizontal range from left to right
Yangle	Overall vertical range from top to bottom

1.25 Test utility TESTJOY

This utility display the current coordinates, radius and angle and the time needed to sample the joystick position. The later determines the timing resolution of MJ and MM.

When starting TESTJOY the current joystick position is interpreted as the center position.

2 New to ERTS V3.32

2.1 Length dependent exposure time for single word presentation

Using the new SETCHARTIME session command you can lengthen the presentation of each single word presented by the SW-trial command depending on the number of characters in the word. For example, if presentation time of a word is defined as 1000ms by the SP-command and the word consists of 6 characters, the overall exposure time would be 1000ms plus 6 x CharTime.

Syntax: SETCHARTIME [CharTime]

Time Extra time for each character in ms (default 0)

2.2 Coding (CE) in trials called by SW-command

The manual describes incorrectly the behavior of CE in combination with CritWord (SW-command).

If CritWord is set to a value greater than 0 and lower than 10000, the CritWord marks one of the words as critical. If a CE command is used in the presentation trial, ERTS will send the first byte of the sequence for all non-critical words and the 2nd byte for the critical.

Using this mode the first byte may be used to reset the port and the second to raise a data line. A trigger will be seen only on the critical word.

```
SEQUENCE OneWord B nnnnnn0 nnnnnn1
```

If the first byte sets one data line and the second another data line, you will see trigger for every word.

```
SEQUENCE EveryWord B nnnnnn1 nnnnnn1n
```

In this case, you need to set ERTSTTLRESET in order to reset each signal automatically, else you will not see any changes on the port when repeated non-critical words are presented (which will be always the case).

If CritWord is set to 10000, ERTS will change in ascending mode and will try to send triggers with each word by selecting bytes in ascending order out of the sequence definition.

```
SEQUENCE EveryWord B
nnnn0001 nnnn0010 nnnn0011 nnnn0100 nnnn0101 nnnn0110 nnnn0111 nnnn1000 nnnn1001
```

2.3 Trial transition matrix for generating constrained trial orders

To generate trial orders which are non-random but rather follow individual transition rules, one or more trial transition matrices can be defined that determine the probability of trial orders. Use numeric SETs to define a trial transition matrix.

The number of rows and columns must correspond to the number of trial specifications of a block definition. If the i-th trial specification of a block has been executed, the i-th row of the matrix will be used to draw the subsequent trial. Each column either includes (1) or excludes (0) trial specification from the set of possible subsequent trials. If, for example, column j in row i contains a 0 then trial specification j will never follow trial specification i. Trial transition matrices can be defined for up to 10 levels which means that you can include or exclude trial specifications depending of what trial specification has been executed ten trials before.

The trial transition mode is activated by inserting a TRANSMAT session command before a RUNBLOCK command.

Syntax: TRANSMAT Mode Mat1 [Mat2 ... Mat10]

Mode Generation mode

- 0: Random selection with replacement from transition matrix
- 1: Random selection without replacement of trial specification. If empty, the set of trial specifications will be resetted automatically. This mode enforces usage of all trial specifications.
- 2: Balanced selection. ERTS will generate a random sequence with equal frequencies for all trials specifications. This mode guarantees that all trials are replicated by the same number.
- 3: Mode 1 and 2

Mat? Name of numeric set which defines a trial transition matrix. Matrix 1 defines probability depending on trial i-1, matrix 10 for i-10

The following example creates a random trial order that follows the following rules: If an odd trial specification (1,3,5,7,9) is executed, the next trial specification will be randomly drawn from the second part of the block definition (6 to 10). If an even trial specification (2,4,6,8,10) is executed, the next trial specification will be randomly drawn from the first half of the block definition (1 to 5). The same trial specification never follows each other.

```
SET TTM1 N
0 1 1 1 1 0 0 0 0 0 ;1
0 0 0 0 0 1 1 1 1 1 ;2
1 1 0 1 1 0 0 0 0 0 ;3
0 0 0 0 0 1 1 1 1 1 ;4
1 1 1 1 0 0 0 0 0 0 ;5
0 0 0 0 0 0 1 1 1 1 ;6
1 1 1 1 1 0 0 0 0 0 ;7
0 0 0 0 0 1 1 0 1 1 ;8
1 1 1 1 1 0 0 0 0 0 ;9
0 0 0 0 0 1 1 1 1 0 ;10
```

BLOCK OddEven 10

```
1 ShowDig 1
1 ShowDig 2
1 ShowDig 3
1 ShowDig 4
1 ShowDig 5
1 ShowDig 6
1 ShowDig 7
1 ShowDig 8
1 ShowDig 9
1 ShowDig 10
```

PARAMETER TMode

```
SESSION Example for constrained random sequences
TRANSMAT 3 TTM1
RUNBLOCK OddEven 100
```

The next example defines two matrices to define the following transition rule: The same trial specification is not repeated within the next 4 trials:

```
SET NoRep N
0 1 1 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 0
```

```
TRANSMAT 3 NoRep NoRep NoRep NoRep
```

3 New to ERTS V3.31

3.1 Direct SoundBlaster Output

To support a larger variety of Laptops and the new SoundBlaster PCI cards, a direct SoundBlaster output has been implemented that does not require pre-installed Creative Labs DOS-drivers for presenting sound samples (VOC, WAV). This new feature solves the following problems

- Missing or bad installation of Creative Labs DOS drivers
- Drivers-related problems on SB compatible cards
- Problems with compatible cards that require enabling routine under DOS. In those cases, loading of DOS drivers in config.sys was impossible. This relates to Laptops and new PCI cards

The new direct SB support is limited to the following sound formats

- 8-Bit mono and 16-bit mono/stereo WAV output (no VOC and no 8-bit stereo)

3.1.1 LS (Load Sound) trial command

The new LS trial command pre-loads sound samples into Extended Memory that are defined in ERTS as a dynamically loadable Tone.

Syntax: LS [ToneName]

ToneName Name of a tone definition of type „S“

Use this new command to control the time when dynamic sound samples are pre-loaded. By default, ERTS preloads tones immediately before presenting them. If the time interval is too short, the LS commands can be used to load samples at the beginning of the trial.

3.1.2 Displaying time of block

Additional place holders in formatted pictures support the display of the elapsed or residual block execution time.

t Elapsed block execution time in seconds
T Residual block execution time in seconds (MAXTIME must be active)

The new LS trial command pre-loads sound samples into Extended Memory that are defined in ERTS as a dynamically loadable Tone.

Syntax: LS [ToneName]

ToneName Name of a tone definition of type „S“

Use this new command to control the time when dynamic sound samples are pre-loaded. By default, ERTS preloads tones immediately before presenting them. If the time interval is too short, the LS commands can be used to load samples at the beginning of the trial.

4 New to ERTS V3.30

4.1 Video support

With ERTS Version 3.30 the FLIC video format is supported with minor exceptions. Videos are defined as PICTURES and loaded just as pictures on screen locations. A new trial command is supported for playing the video(s) loaded with SP or FP commands.

4.1.1 New PICTURE type 'V'

To define a Video use the PICTURE command with the new picture type 'V'. Specify the full path and filename to load a FLIC (FLI or FLC) files. there are several new subtypes that may be used to specify certain variants of videos. FLIC files are played directly from disk. The first frame is internally preloaded into the invisible screen buffer and displayed with one screen refresh. All subsequent frames are rendered into the visible screen buffer. Video pacing can be done by either millisecond times or by multiples of screen refreshes. For the later the nearest multiple of screen refreshes are used to reflect the frame rate as defined in the FLIC file.

Syntax: PICTURE name V[subtype] file

subtype

- r Repeat Video. Note that the file must include a ring frame that defines the transition between last frame and second frame. This is the default for FLIC.
- s Single frame. Only one single frame is displayed and the video will not run.
- m Load file into conventional memory and play from memory. Note that this is only recommended for small videos. (see also subtype C)
- c Copy video definition and share memory. If you use this subtype specify a name of a picture instead of a file name. ERTS does copy the picture definitions and shares the video data without allocating new memory. Use this subtype to create copies of the same video if you want to display multiple instances of the same video at the same time (e.g. as distractors). Note that you can only copy pictures that have been defined with subtype 'm'.

4.1.2 New PV (Play Video) trial command

The PV trial command plays all videos that have been loaded with previous SP or FP commands. If you insert a WK command between the SP and PV command, the first frame of a video will be visible and the subject can start the video by pressing a key.

Syntax: PV [MaxTime [Cancel [NFrames [Continue [Speed [FrameSync]]]]]]

Time	Max duration of video sequence
Cancel	Cancel mode
NFrames	Number of frames to be played (use this to run only a subset of the video)
Continue	0: Unload all videos after PV is finished. 1: Hold all videos for a second PV-command. The default is 0. Use this argument if you want to interrupt a video or do some intermittent activities.
Speed	Speed in millisecond that overwrites default speed. Default is 0 (use default speed)
FrameSync	Play video by using constant multiples of screen refresh periods instead of millisecond timing. This mode is recommended if you want to exactly sync the frame rate of two parallel videos or if you prefer to display a video synchronized to the screen refresh instead of realizing the exact video speed.

4.1.3 New VA (Video Attributes) trial command

The VA trial command may be used to modify and extend the attributes of a single video.

Syntax: VA [PicName [Speed [StrtFrame [CritFrame [Key [Tone [SequOn SequOff Port]]]]]]]]

PicName	Name of a Picture definition of type video
Speed	Sets new video speed. A 0 means use speed as defined in file
StrtFrame	Sets new startframe for video. A 0 means start with 1 frame
CritFrame	Defines a certain frame as the critical frame. Specific events can be triggered with this critical frame. (see next arguments)
Key	Generate internal response using this key if critical frame is displayed. Use this option similar to the GR trial command to generate an internal response as a time stamp. Note that you must have activated response registration for at least 2 responses.
Tone	Play this tone if the critical frame is displayed. Do not use tone definitions that require long loading times
SequOn	Send this sequence to the parallel port if the critical frame is displayed
SequOff	Send this sequence to the parallel port if the frame after the critical frame is displayed
Port	Use this port for SequOn and SequOff

4.1.4 Testing videos with option /Tp

The /Tp-command line option (Test Picture) does support testing of videos. By default, the first frame is displayed including the time required to render this frame. If you press the 'P'-key the video will be played. The max time required to render a frame is displayed.

4.1.5 Examples

Playing a single Video

```
PICTURE myVideo V video.flc
POSITION Center P 0,0
TRIAL Play
SP myVideo Center
WK
PV
```

Playing a looping Video until key is pressed

```
PICTURE myVideo Vr video.flc
POSITION Center P 0,0
TRIAL Play
SP myVideo Center
RK
PV 0 1
```

Visual search using small videos

```

PICTURE Target Vm target.flc
PICTURE Dist1 Vm distract.flc
PICTURE Dist2 Vc Dist1
PICTURE Dist3 Vc Dist1
PICTURE Dist4 Vc Dist1
PICTURE Dist5 Vc Dist1
PICTURE Dist6 Vc Dist1
PICTURE Dist7 Vc Dist1
PICTURE Dist8 Vc Dist1

SET Distractors V Dist1 Dist2 Dist3 Dist4 Dist5 Dist6 Dist7 Dist8

POSITION Circle P 0,0 800 0 8

TRIAL Play
MP Circle
RK
FP ? Circle 1 0 Distractors 0 1
PV 0 1

BLOCK Vsearch 2
1 Play Target
1 Play Dist1

```

5 New to ERTS V3.29

5.1 General ERTS commands

5.1.1 Parameter Values in command line

The `PARAMETER` command without a value definition lets you specify values when you start an experimental session. You may now add values in the command line as well to support batch execution mode:

Syntax: `&value`

The ‘&’-symbol marks command line arguments as parameter values. The order must correspond to the `PARAMETER` commands in the script.

5.1.2 TONE command supports WAV-format

The SoundBlaster support has been extended to the WAV-file format. You must include the file extension WAV when defining a tone. To play WAV-files, the Creative Labs DOS drivers `CTWMEM.DRV` and `CTWDSK.DRV` must be present.

5.1.3 Extended CS-command for equi-distant random time interval

In addition to the exponential random time interval, the `CS`-command now supports the definition of an equi-distant random time interval. Whenever `MaxRnd` is less or equal to `RndTime`, ERTS will interpret these values as follows:

`RndTime` Time range that should be varied at random

`MaxRnd` Number of different random time intervals within this range

For example, if `MaxRnd=5` and `RndTime=4000` then five different equi-distant time intervals are drawn at random: 0,1000,2000,3000,4000.

5.2 External Counter (EEG-Version only)

Typical MRI-settings require that trial and block execution is synchronized by the scan periods of the MRI. To support such a „scan-paced“ execution mode a new set of commands is supported in the EEG-version that let you define a counter which is based on external tics. These external tics will usually be TTL scan-signals from the MRI which are integrated via the EXKEY interface.

5.2.1 EXCOUNTER session command

This command defines the external counter by defining a key input as the external tic. If you define one of the EXKEY keys (E1 – E16) as the counter tic you can define an external TTL-signal as the source of the counter.

Syntax: EXCOUNTER [key]

Key One of the pre-defined key names (default is no key)

5.2.2 MAXCOUNT session command

The MAXCOUNT session command defines a maximum block or trial execution time in terms of external tics. Use this command to limit the duration of a trial or block to a certain number of scan periods.

Syntax: MAXCOUNT [MaxBlk [MaxTrl]]

MaxBlk Number of tics after which a running block should be aborted (default is 0)

MaxTrl Number of tics after which a running trial should be aborted (default is 0)

5.2.3 WC trial command

The WC-trial command (Wait Count) waits for a certain number of external tics. Use this command to synchronize trial events with external events

Syntax: WC [Ntics [Reset]]

Ntics Number of tics to wait (default is 0)

Reset Reset counter after waiting

0 Do not reset

1 Reset to Null (default)

By using the Reset argument you can decide whether you want to use the external counter in absolute mode or in lap mode. Note that resetting this counter does not influence the MaxBlk and MaxTrl abort conditions set by MAXCOUNT or MC.

5.2.4 MC trial command

The MC-trial command (Max Count) limits the execution time of a trial to a certain number of tics. This is equivalent to the second argument of the MAXCOUNT session command except that this command is used on the trial level and thus enables you to define different values for trials within one block.

Syntax: MC [MaxTics]

MaxTics Number of tics after which this trial should be aborted (default is 0)

This command counts always from the beginning of a trial independent where you place this command within the trial definition. However, the command will not be active before it is executed. Therefore, you should position it early enough within the trial sequence.

5.3 Serial Output (EEG-Version only)

The output channel of the SS-trial command and of all EEG-syncing commands can now be defined as serial output channels. The channel numbers 1-4 that are specified as an argument in the trial or session commands usually correspond to the LPT1-4. You can overwrite this assignment in the ERTS.CFG file using the ERTSCFG configuration program. To define one of the channels as a serial output channel a new SERIALPORT command has been implemented.

5.3.1 SERIALPORT

The SERIALPORT command is a global system command which is usually placed at near the beginning of a script file. It assigns one of the four output channels to one of the serial COM ports of your computer.

Syntax: SERIALPORT Channel COM# Baud Bit StopBit Parity DTR

Channel	Number of channel (1-4) to be used for serial output.
COM#	Number of COM port (1-3)
Baud	Baud rate (2400, 4800, 9600, ...)
Bit Format	(5,6,7,8)
StopBit	(1,2,)
Parity	0: None, 1:Odd, 2:Even
DTR	Use Data Ready Flag (0: No, 1:Yes)

5.4 External definition file for key names of PC keyboard

The table of pre-defined ERTS key names has been separated from the ERTS program to allow users to localize key names for different keyboard layouts and to overwrite key names for assigning special screen echo characters to keyboard keys.

The standard name of the key definition file is ERTS.KEY. This key must be present in either the ERTS system directory or the current directory. To activate other keyboard layouts either overwrite this standard file or add the name of another key definition file to the command line when calling ERTS. This file must have the file extension .KEY in order to be recognized by ERTS as a key definition file. Look in your ERTS system directory for prototypes of keyboard files for different countries. Their name is KEY.nnn where nnn is the country.

Each line in the key definition file assigns one key name to an internal keyboard scan code

ScanCode : KeyName

See the standard keyboard file for the meaning of the scancodes.

For overwriting key names you may create partial definition files which only assign new names to task relevant scancodes. You must not define duplicate names.

The following keyboard file assigns dummy names to the digit keys 1-4 to make the names available and then assigns the key names to the keyboard keys A, X, M, L. Now you could use these keys as response keys which produce the symbols 1,2,3,4 as screen echo.

```
2 : D1
3 : D2
4 : D3
5 : D4
```

```
30 : 1
45 : 2
50 : 3
38 : 4
```

5.5 SHOWTTL utility

With the SHOWTTL utility you may debug and verify the correct sending of ERTS sync signals. For this you must connect your ERTS computers via parallel cable (e.g. LapLink cable) with another PC. While running an ERTS script which includes sync signals, you may run SHOWTTL on the other (receiving) computer and watch the timing of incoming signals in millisecond resolution. Note that you must run SHOWTTL not within a MS-DOS box in order to obtain accurate measurements.

```
SHOWTTL port [/T] [/W byte]
```

```
Port   Number of parallel port (LPT 1,2,3,4)
/T     Show time markers only. Use /T if screen updating of SHOWTTL takes longer than
       intervals between signals.
/W     Write byte value (Dec) to parallel port. Use this option to write (send) a byte to a
       parallel port e.g. to test the connections between the two computers or to reset a port.
```

5.6 ERTSCODE

5.6.1 EXCLUDE

A new EXCLUDE command allows filtering of values that are out of a range whereas the range is computed by using all data. The command functions similar to the SKIP command but causes ERTSCODE to process each file in two phases. In phase one, the EXCLUDE command is not executed, thus all data are processed and all statistics are computed based on the entire data set. In phase two, the EXCLUDE command is processed, thus data that do not meet the WHERE condition are excluded.

Syntax: EXCLUDE WHERE condition

Note that in order to function as a two phase exclusion criteria you must compute variables based on the statistical analysis of the first phase. This is typically done by using END_COMPUTE statements to set the variables:

```
NUMERIC Mini
NUMERIC Maxi

EXCLUDE WHERE num(R1) GT Maxi OR num(R1) LT Mini

...

END_COMPUTE Mini = CELL("Mean") - CELL("STDV")
END_COMPUTE Maxi = CELL("Mean") + CELL("STDV")
```

5.6.2 Extended /E (Export) command line option

The command line option /E (re-directing SHOW output to a file) has been extended to support RTF-file format and the appending of new output to existing files.

Syntax: /E[WriteMode[:rtf]]

The WriteMode is one of the symbols ':', '*', '+', '='.

```
: Normal (Overwrite existing files)
* First (Overwrite existing files, do not close rtf-file)
+ Continued (Append to existing file, do not close rtf file)
= End (Append to existing file, close rtf-file)
```


Adding the string „rtf“ will enable RTF-output format. For example

```
/E:rtf
```

will activate RTF-format. For non-rtf format only WriteModes ‘:’ and ‘+’ make sense.

6 New to ERTS V3.27

6.1 General ERTS commands

6.1.1 Simultaneous use of input devices

ERTS internal architecture of collecting responses has been changed to support the collection of responses from different input devices at the same time. Consequently, KEY definitions may now consist of key names that are a mixture of different input sources.

6.1.2 UK (Update Key Definition) trial command

If the correct response sequence depends on a randomly generated order, this command may be used to derive the updated order of keys from a set of key definitions.

UK [KeySet KeySequ NKeys Reverse]

KeySet	Set consisting of key definitions that will be written into KeySequ
KeySequ	Key definition that will be updated according KeySet
NKeys	Number of keys to be copied from KeySet
Reverse	Determines order of copying
0	Normal order (default)
1	Reverse order

7 New to ERTS V3.26

7.1 General ERTS commands

7.1.1 External index files for controlling trial order

The RUNBLOCK command has been extended to support external index files. ERTS will execute a block definition according to this external file.

```
RUNBLOCK      N(Valid)|IndFile(Valid)      N(WarmUp)|IndFile(WarmUp)      RndMode(Valid)
RndMode(WarmUp) RepMode(Valid) RepMode(WarmUp)
```

Instead of defining the number of trials you may refer to an index file. This file must list the overall number of trials in the first line. The following lines list the indices one per line. An index selects the i-th trial specification from within a block.

7.1.2 Modified SW-trial command

The SW-command has been extended by the argument „CritWord“. This is the index of the critical word within the picture that is presented in SingleWord-Mode. With this new argument you may define specific sync-triggers and generated responses for the critical word only.

SW TextPic MaxWords CritWord SWmode SWmask SWtrial VarArgList

CritWords Index of critical word. If this is > 0, the behavior of the GR and CE command will be modified.

7.1.3 GR (Generate Response) trial command

A new trial command can be used to generate an artificial response synchronous to the next trial event

GR Key

Key Name of a pre-defined key that will be registered as pressed

This trial command helps you to automatize computation of time differences between a subject's response and a stimulus onset. Whenever you have the response registration turned on before a stimulus onset occurs, this trial command will help you to determine the exact time difference between stimulus onset and response. For example, if you ask a subject to synchronize the key press with the onset of a second stimulus (estimate time interval), you want to record responses before and after the second stimulus in order to get early and late responses. If you generate an artificial response together with the second stimulus, you can calculate the lap between onset and response by subtracting the latency of the real reaction from the artificial reaction.

7.1.4 CJ (Conditional jump) trial command

This new trial command can be used to branch to another trial specification within a block definition or to a session label within the session definition dependent on feedback classifications. TrialIDs or session labels are used as destination labels.

CJ Label1 Label2 Label3 ... Label12

Label? Destination label that should be branched to on the occasion of feedback class ?.
Use the TrialID to branch to a certain trial within the block definition, use -N or +N to make a relative jump N-th trials forward or backward, use a session label to end a block and to jump to a session label, or use the reserved name „STOP“ to finish a block and continue the session definition.

Note that the randomization mode must be 0 when you use absolute trial branching and 0 or 3 if you use relative trial branching.

7.1.5 AP (Align Picture) trial command

This trial command may be used to change the default centering on screen locations when displaying pictures.

AP [Mode]

Mode Determines alignment of pictures
0 Centered (default and ERTS standard)
1 Left aligned
2 Right aligned

- 3 Alignment depends on X-coordinate
 - Right, if in left half of screen
 - Left, if in right half of screen
 - Centered, if in horizontal middle of screen

You may concatenate the AL command with other SP and SF trial commands to load pictures with different alignments as one complex image

7.2 StairCases

The implementation of staircases within ERTS is not simple because ERTS does not support computation of variables. Instead, staircases in ERTS recalculate variable argument values of a block definitions. The up/down process is based on feedback classifications.

To achieve this, you must follow these steps:

1. Define a staircase with the proper basic characteristics
2. Define a trial with the staircase relevant parameter as a variable parameter
3. Add a feedback command that classifies the response.
4. Initialize the starting values of the staircase by using the INITSCASE command.
5. Link the staircase the a variable parameter column by using the USESCASE-session command

If a parameter is not a numeric value like time or PosIndex but rather represents an array of names (several pictures of different sizes) then create a SET consisting of those objects and use the index argument of the DE trial command to select the i-th element out of the set.

7.2.1 New STAIRCASE Data object

For running staircases within a block, you must first define a staircase object which holds basic parameters of your staircase process such as the start value and UpDown characteristics.

STAIRCASE Name StartAfterTurn EndOnTurn DivOnTurn DivByTrial RepeatDown RepeatUp
AverageTurns AutoReset IndexFile

Name Name of new staircase object

StartAfterTurn Number of turns until internal averaging begins. A null starts averaging immediately. Note that this parameter is only relevant if you use the SETSCASE or FIXSCASE session commands.

EndOnTurn Index of turn at which a running block should be suspended. Use this parameter to abort a block depending on the number of turns of your UpDown process. A block will not be aborted until all running Staircases are suspended.

DivOnTurn Number of turns at which step values will be divided by two. A null disables division.

DivByTrial Determines whether step sizes should be divided by trial number.
0 No
1 Yes, divide by number of trials. Note that if DivOnTurn is also active, DivByTrial will start with $N(Trl) = 1$ after DivOnTurn ends.

RepeatDown/Up	Number of consecutive repetitions until the UpDown process reacts. (Default is 1)
AverageTurns	Use turns only to estimate average value. Note that this parameter is only relevant if you use the SETSCASE or FIXSCASE session commands.
AutoReset	Enables auto resetting of staircase <ul style="list-style-type: none"> 0 Do not reset staircase 1 Reset staircase after each block 2 Reset staircase after each trial
IndexFile	Optional path to a an index file that contains an array of values. The staircase's value will be used as an index into this array. This way you may translate constant step sizes by another functions (e.g. psychometric function). Format of index file is number of values in first line followed by all values one per line.

7.2.2 INITSCASE session command

This command initializes a staircase with starting values. You must use this command first before using USESCASE.

INITSCASE Name StrtVal MinVal MaxVal StepDown StepUp UseRetrace

Name	Name of staircase
StartVal	Start value of staircase <ul style="list-style-type: none"> > 0 Value to be used on first iteration. If UseRetrace is active this initial time value will be rounded to the best matching retrace time. 0 MinVal or MaxVal will be used at random as start value < -1 Random start value between MinVal and MaxVal
Min/MaxVal	Minimum/Maximum Value that may be reached during UpDown process
StepDown/Up	Step size to go down/Up. If UseRetrace is active this is the number of retraces to decrease/increase value.
UseRetrace	If 1 then StepsUp and StepDown will be interpreted as multiples of one retrace period. The Startvalue will be rounded to the closest multiple of refresh periods.

7.2.3 USESCASE session comand

This command activates a staircase and assigns it to a variable argument column of a block definition. You must have initialized the staircase with INITSCASE before using it.

USESCASE Name VarArgIndex [BlkStart [BlkStep [BlkEnd [F1..F12]]]

Name	Name of staircase to use
VarArgIndex	<ul style="list-style-type: none"> > 0 Index of the variable parameter column. Note that this requires a homogenous structure with respect to variable arguments across trials within one block definition. 0 The staircase values will be intepreted as a line index into a block definition. Thus, the staircase will walk through your block definition instead of varying the value of one variable argument.

BlkStart/Step/End If these values are not null the staircase will only be used for trial specifications beginning with **BlkStart**, incrementing in **BlkSteps** and ending at **BlkEnd**. Use these arguments to limit the staircase to a subset of trial specifications within a block.

F1..F12 These optional arguments can be used to overwrite the default actions of the feedback classes. By default, only class 2 (correct response) walks down and all other walk up. You may define one of the following actions for each feedback class:

- 1 Walk up
- 1 Walk down

7.2.4 SETSCASE session comand

This command assigns a constant value to a variable argument column of a block definition. The value is derived by the staircase's current mean and some delta. You can use this session command to create a set of constant values for a subsequent block centered around a staircase's mean. In other words, this command enables you to adjust the method of constant stimuli individually by running a staircase block first.

SETSCASE Name Delta Block VarArgIndex [BlkStart [BlkStep [BlkEnd]]]

Name Name of staircase to use

Delta A Delta value which is added to the staircase's mean

Block Name of the block definition where values should be written to

VarArgIndex Index of variable argument column within the block

BlkStart/Step/End If these values are not null the values will only be written to trial specifications beginning with **BlkStart**, incrementing in **BlkSteps** and ending at **BlkEnd**.

7.2.5 STOPSCASE session comand

This command disable an active staircase that has been enabled by a **USESCASE** command.

STOPSCASE Name

Name Name of staircase to stop

7.2.6 FIXSCASE session comand

This command fixates the value of an active staircase according to its internal estimate. The state of other active staircase's might be considered as well. This staircase can be still active, but the UpDown process will be stopped.

SETSCASE Name1 Name2 [Name3 [Name4]]

Name1 Name of staircase to fixate

Name2/3/4 Name of staircase to use as additional source for averaging

7.3 ERTS Font Converter

A new ERTS Font converter is included in the sub-directory /EFOCONV on the Sample Disk. This font converter is a Windows application that lets you convert and export any TrueType-Font into a font resource file that can be used from within ERTS.

The font converter requires windows and can be install by executing SETUP.EXE in the sub-directory /EFOCONV on the sample disk from within windows.

This font converter is shipped on an „as is“ base. Note that any existing copyrights concerning windows fonts must not be violated by the end-user.

Note that for each font size and type you must create a single font resource file. To save disk and memory space you may select a sub-set of characters to be included in the font resource file.

Before generating a font you must also set the characteristics of your destined graphics mode / monitor in order to get a properly sized raster font within ERTS.

To use the new font resource files within ERTS you must append the extension .EFO when loading the font with the FONT command.

8 New to ERTS V3.25

8.1 General ERTS commands

8.1.1 Sine-distributed positions

The position definitions of type 'line' and 'vector' have been extended to support sine-distributed positions for moving pictures back and forth at sine-wave speed:

```
POSITION name L xs ys xe ye Npos [Sine]
```

or

```
POSITION name V x y Length NPos Angle [Sine]
```

```
Sine    Determines spreading of positions on line
        0    Equally spaced
        i > 0 Sine-distributed
```

8.1.2 Embedded color commands within text picture

The picture type 'F' (Feedback picture) has been extended to support embedded color format fields using the similar notation as the embedded feedback fields. For changing foreground color within the picture use the following syntax:

```
Set Color:  @Ccolor@
Reset Color: @C@
```

The 'C' is used to differentiate between embedded color format field uniquely and feedback fields. For setting a new foreground color append the name of the color definition and a trailing @-symbol.

8.2 New EEG-Version of ERTS

Since ERTS version 3.25 EEG-syncing has been improved by adding new commands, but at the same time EEG-syncing has been removed from the standard version of ERTS and ERTSVIPL. EEG-syncing is now only available with the license extension for EEG!

With earlier versions of ERTS a sync signal had to be defined by inserting an SS command immediately after the corresponding presentation command. In order to execute this SS command virtually simultaneous with the stimulus presentation, the time of the presentation command had to be set to Null and possible intermediate pre-loading activities had to be ruled out by setting a LB command after the SS.

The new commands simplify this procedure.

8.2.1 CE (Code Event) trial command

The CE-trial command defines a single byte pattern that will be sent to the parallel port at the time the next visual or auditory event starts. As the sync signal will be sent automatically with the following event, the signal cannot be delayed by internal pre-loading of images.

CE sequence [Channel]

Sequence	Name of a byte sequence containing one single byte
Channel	Number of output channel (LPT 1,2,3,4)

The byte will be directly written to the port without any handshake (this corresponds to Bios mode 1 in the SS command)

```
CE StimOn
RK ResKEys
SP ? Center 1 0 1
```

8.2.2 CODEEVENTS session command

This session command lets you define automatic sync signals for certain sub-sets of trial events on the session level. Using this session command you do not need to alter your trial definitions:

CODEEVENTS Channel [ResetSequ [RtSequ [FBSequ [VisualSequ [AudiSequ [ParSequ]]]]]]

Channel	Number of LPT output channel (default 1).
ResetSequ	Name of a sequence that is sent at the beginning of each trial.
RtSequ	Name of a sequence that is sent at the beginning of response registration
FBSequ	Name of a sequence that contains several bytes which are sent for each FB class
VisualSequ	Name of a sequence that contains several bytes which are sent with the i-th visual event
AudiSequ	Name of a sequence that contains several bytes which are sent with the i-th auditory event
ParSequ	Name of a sequence that contains several bytes which are sent with the i-th parallel output.

All of these arguments can be set to NULL to use only a subset of these automatic event coding mechanisms. The Reset and RtSequ must consist of only one single byte. The FBSequ may consist of several bytes. From those, a corresponding byte is selected according to the feedback class. The Visual, Auditory, and Par sequences may also consist of more than one byte. If a trial consists of more than one event of a given modality, then the bytes will be selected in ascending order. If less bytes are defined than events, the highest available byte will be used.

```

SEQUENCE Reset B 00000000
SEQUENCE RtSequ B nnnnnnn1
SEQUENCE FBSequ B nnnn001n nnnn010n nnnn011n nnnn100n

SESSION
CODEEVENTS 1 Reset RtSequ FBSequ

```

In this example a Null byte would be sent to port at the beginning of each trial. With each event following a RK-command (Register Key) the RtSequ sequence would be sent. Each FB-command would also sent one of the four FBSequ bytes depending of the feedback classification (1-4).

8.2.3 CODEKEYS (Modified KEYCODE command)

In order to provide an homogenous set of commands the KEYCODE command has been modified to match the syntax of the new CODEEVENTS command. The channel argument now comes first and there is no Bios argument for setting the handshake. CODEKEYS does now only support direct writing.

Syntax: CODEKEYS [Channel KeySet [ByteSequ [ToneSet]]]

Channel	Channel number 1-4 (translated by ERTSCFG into parallel port numbers). Default 1
KeySet	Name of a key definition which contains keys to be coded.
ByteSequ	Name of a byte sequence which contains bytes for each of the keys. (default none)
ToneSet	Name of a set which contains tone definitions for each of the keys. (default none)

8.2.4 Strobe like sync signals

The new CE and CODEEVENT commands will write a byte directly to the parallel port. In some cases it might be useful to turn these signals into short strobe signals by resetting the port after a short delay. This may be achieved by setting an appropriate ERTSTTLRESET DOS environment entry.

SET ERTSTTLRESET=Delay[,Invert]

Delay	Length of Strobe in Microseconds (! No millisecond). A Null byte will be send to reset the port
Invert	This argument may be used to invert all bit patterns. 0 Use NULL to reset the strobe signal (default) 1 Invert all (!) Sequence definitions and use FF hex to reset the strobe signal.

Note that all parallel port output including the signals from CODEKEYS and SS commands are converted into a strobe signal.

8.2.5 SHOWTTL utility

With the SHOWTTL utility you may debug and verify the correct sending of ERTS sync signals. For this you must connect your ERTS computers via parallel cable (e.g. LapLink cable) with another PC. While running an ERTS script which includes sync signals, you may run SHOWTTL on the other (receiving) computer and watch the timing of incoming signals in millisecond resolution. Note that you must run SHOWTTL not within a MS-DOS box in order to obtain accurate measurements.

SHOWTTL port [/T] [/W byte]

Port	Number of parallel port (LPT 1,2,3,4)
/T	Show time markers only. Use /T if screen updating of SHOWTTL takes longer than intervals between signals.
/W	Write byte value (Dec) to parallel port. Use this option to write (send) a byte to a parallel port e.g. to test the connections between the two computers or to reset a port.

8.3 ERTSCODE

8.3.1 NUMERIC keyword for declaring numeric variables

The implicit declaration of numeric variables has been removed from ERTSCODE to provide better checking of ERSTCODE-jobs. You must now declare each numeric variable at the top of a definition file by using the new NUMERIC keyword:

Syntax: NUMERIC VarName

VarName Any user-defined name for this variable

8.3.2 IMPLICIT keyword

The old-style implicit declaration of numeric variables can be enabled by inserting the new IMPLICIT at the top of a definition file:

Syntax: IMPLICIT

Note that allowing implicit declaration has the potential danger of not recognizing misspelled variable names or non-existing columns in the result file because ERTSCODE interprets them implicitly as new variables.

8.3.3 New Start_ and _Final compute blocks

The concept of executing BEGIN and END compute blocks has been extended by START_ and FINAL_ to improve handling of multiple input files.

START_	is executed in the first data row of the first file before BEGIN and normal block
BEGIN_	is executed in the first data row of each file before normal block
END_	is executed in the last data row of each file after normal block
FINAL_	is executed in the last data row of the last file after normal and END block

8.3.4 SHOWLIST

A new SHOWLIST command simplifies the creation of output tables that list data from multiple files. Similar to the OUTPUT command you may list several text variables which will be included in the output list as columns. Global delimiter and justification settings are used.

Syntax: SHOWLIST varname1 [varname2 ... varnameN]

ERTSCODE produces one label line and then one output line per file. Using the /E option you may redirect this output into a file. In contrast to the OUTPUT command you are completely free to arrange and include certain statistical parameters. To extract parameters from an factorial analysis, declare a text variable and assign the value to this variable within an END_block:

```
COLUMN MeanRt 10
```

```
END_COMPUTE MeanRt = CELL(„Mean“, „Response.Correct“)
```

8.4 Incompatibilities to older ERTS versions

- New DROPOUT session command substitutes the Block randomization modes 4 and 5. These modes are not supported anymore

- The meaning of the 'Sequence' argument of the FB-command has been slightly changed.
- The Level=1 argument of the CC-trial command (Change color) now manipulates the first loaded foreground color and not necessarily the standard foreground color.
- The KEYCODE session command is changed to CODEKEYS. Syntax has been changed and only direct writing to port is supported (corresponds to Bios=1).
- Since ERTSCODE V3.08 numeric variables must be declared explicitly by using the new NUMERIC keyword. To use old-style implicit declaration insert the new keyword IMPLICIT at the top of the ERTSCODE-definition file.

9 New to ERTS V3.24

9.1 General ERTS commands

9.1.1 Extended SETMSEARCH command for presenting any sequence of stimuli as dual task to tracking

The SETMSEARCH and RUNMSEARCH commands have been extended to support the presentation of list of single stimuli. There are two new arguments ForceEndOfTracking and ListMode.

ForceEndOfTracking control whether tracking time will be limited by length of stimulus list.

- 0 No (default)
- 1 Yes, tracking will stop after last item of dualtask has been presented.
Note that tracking time must be sufficient large to finish list, but not too long because result areas are allocated for full length

ListMode Determines presentation type of Target and Probe stimulus sets

- 0 Memory Search: Use arguments TargSet as target and ProbeSet as probe sets for Sternberg memory search (default)
- 1 Random list: Present all items of ProbeSet in random order
- 2 Fixed List: Present all items of ProbeSet in fixed order

In ListModes > 0 you may use the argument TargSet for coding items as positive items, i.e. where response yes is correct. You must enumerate all items from ProbeSet which are positive items in the TargSet set definition. TargSet may be set to NULL by setting a hyphen (-) instead of a name to disable response classification.

9.1.2 New support for mouse/touchpad click-responses

The new TP (TransPos) trial or session command enables you to define click-sensitive areas on the screen and to assign response-key names to those screen areas.

TP [Position KeyTbl Radius Height [Cursor [StrtPos] [ClickMode] [ClickPic]]]

Position Position definition which defines the center of sensitive screen locations

KeyTbl	Key definition which is used as the translation table between screen location and response key name. A click on a position with index <i>i</i> will be coded with the <i>i</i> -th key name within the key table.
Radius	Radius of the sensitive areas in mm/10. If the next is argument (Height) is $< > 0$ then this argument will interpreted as horizontal width to describe a rectangular area instead of a circle.
Height	Height of sensitive screen area. This is optional and may be set to 0 to define circle like areas (see previous argument Radius).
Cursor	Optional set definition with one or more picture elements who are used as bitmaps for the mouse screen cursor. The first picture defines the bitmap to signal off-target status and the next picture(s) to signal on-target status. You must only use picture types which are simple 2-color bitmaps, e.g. Pixel, GRA, Rotated pictures). If more than two pictures are in the set of pictures, the <i>i</i> +1-th picture will be used to signal on-target status on the <i>i</i> -th position. If you use a touch pad as input device it makes no sense to define a screen cursor.
StrtPos	Position definition which sets the start location of the mouse cursor.
ClickMode	Sets visual feedback to click responses (recommended for multiple response registration ($N(\text{Key}) > 1$) <ul style="list-style-type: none"> 0 No feedback (default) 1 Inverse sensitive area while button/finger is down 2 Clear sensitive area
ClickPic	Optional set definition with one or more simple 2-color bitmap pictures which will be displayed as feedback on the corresponding position. If more than one picture is in the set of picture, the <i>i</i> -th picture will be used to signal on-target status on the <i>i</i> -th position

To disable the translation of positions use TP trial command or TRANSPOS session command without arguments.

You may use the new OK trial command to define visual feedback for each response location (key name)

9.1.3 New OK-trial command (OnKey) for key-dependent feedback

As an extension to the FB-trial command which allows response-dependent feedback to categories such as correct/false, the OK-trial command supports the definition of feedback to particular response keys.

OK [Sequential KeyIndex]

Key1: Trialcommand

Key2: Trialcommand

KeyN: Trialcommand

Sequential Determines the way multiple responses are compared

0 Look for the key among all key presses (default)

1 Interpret multiple key presses as a input sequence and compare with key sequence within Key?.

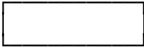
KeyIndex Selects a single key press within multiple responses for comparison

0 Compare all key presses (default)

>0 Compare *i*-th key press

Key? Name of predefined key or name of key definition which is matched against key presses to trigger execution of trial command.

The following example demonstrates how the new TransPos command and the OK command may be used to define a point&click response mode with sensitive key symbols on the screen that flashes in response to a click.

RETRACE 1**COLOR** Blue C 240 100 100**COLOR** Gray C 0 0 50**COLOR** White C 0 0 100**SCREENCOLOR** White Gray**PICTURE** Boris N**PICTURE** Becker N**PICTURE** Steffi N**PICTURE** Graf N**PICTURE** Michael N**PICTURE** Jackson N**PICTURE** Whitney N**PICTURE** Houston N**PICTURE** KeyUp T**PICTURE** KeyDown T**PICTURE** OnTarget P

00001100000

00001100000

00001100000

00001100000

11111111111

11111111111

00001100000

00001100000

00001100000

00001100000

PICTURE OffTarget P

11111111111

11111111111

1100000011

1100000011

1100000011

1100000011

1100000011

1100000011

11111111111

11111111111

SET Cursor V

OffTarget OnTarget

KEY KeyTbl 1 2 3 4**POSITION** KeyPos M 0,500 500 500 4 1**POSITION** StartPos P 0, 200**POSITION** Center P 0,0**TRIAL** ShowNames**CS** 2000**RK** ML**SP** Becker KeyPos 1**&SP** Graf KeyPos 2**&SP** Jackson KeyPos 3**&SP** Houston KeyPos 4**&SP** KeyUp KeyPos 1 0 0 Blue**&SP** KeyUp KeyPos 2 0 0 Blue**&SP** KeyUp KeyPos 3 0 0 Blue**&SP** KeyUp KeyPos 4 0 0 Blue**&SP** ? Center 1 0 1**FB** ?**+ OK**1:**SP** KeyDown KeyPos 1 500 0 Blue2:**SP** KeyDown KeyPos 2 500 0 Blue3:**SP** KeyDown KeyPos 3 500 0 Blue4:**SP** KeyDown KeyPos 4 500 0 Blue

```

BLOCK Test 4
1 ShowName Boris 1
1 ShowName Steffi 2
1 ShowName Michael 3
1 ShowName Whitney 4

```

```
SESSION Test new TP command (Transform Positions)
```

```
TRANSPOS KeyPos KeyTbl 400 200 Cursor StartPos
```

```
RUNBLOCK Test
```

9.1.4 Extended image to image transfer capabilities

The TI (Transfer Image) trial command has been extended to support transfer in both directions between current image and additional images, and to support masked transfer of images for easy masking or degrading of color stimuli.

Syntax: TI DestImage SrcImage Masked

DestImage	Index of destination image, 0 means default IPL-page
SrcImage	Index of source image whose content should be transfer to the destination image 0 means default IPL-page
Masked	Flag that determines the transfer mode 0 Normal transfer (default) 1 Masked transfer: The source image is only transferred to non-background areas of the destination image.

then load pictures into the default IPL-page to define the transparent areas for the image transfer process, and then transfer the image from the additional screen image into the default IPL-page with the masked-flag set to On.

To mask a color picture you must first load it into an additional screen image,

```

TRIAL LoadImage
LI 1
SP Pic Center
LI

```

then load one or more pictures as the mask into the default IPL-page to define the transparent areas for the image transfer process and then transfer the image from the additional screen image into the default IPL-page with the masked-flag set to On. Currently you must trigger the display of the new masked image by using an extra ShowPic command with an empty dummy picture.

```

TRIAL MaskImage
SP Mask MaskPos
&TI 0 1 1
&SP EmptyPic Center 1
WK

```

9.2 External definition file for key names of PC keyboard

The table of pre-defined ERTS key names has been separated from the ERTS program to allow users to localize key names for different keyboard layouts and to overwrite key names for assigning special screen echo characters to keyboard keys.

The standard name of the key definition file is ERTS.KEY. This key must be present in either the ERTS system directory or the current directory. To activate other keyboard layouts either overwrite this standard file or add the name of another key definition file to the command line when calling ERTS. This file must have the file extension .KEY in order to be recognized by ERTS as a key definition file. Look in your ERTS system directory for prototypes of keyboard files for different countries. Their name is KEY.nnn where nnn is the country.

Each line in the key definition file assigns one key name to an internal keyboard scan code

ScanCode : KeyName

See the standard keyboard file for the meaning of the scancodes.

For overwriting key names you may create partial definition files which only assign new names to task relevant scancodes. You must not define duplicate names.

The following keyboard file assigns dummy names to the digit keys 1-4 to make the names available and then assigns the digit names to the keyboard keys A, X, M, L. Now you could use these keys as response keys which produce the symbols 1,2,3,4 as screen echo.

```
2 : D1
3 : D2
4 : D3
5 : D4
30 : 1
45 : 2
50 : 3
38 : 4
```

9.3 ERTS and MS-Windows 95

Win95 provides much better support for activating DOS-applications from within Windows than Win 3.x. You can start DOS-applications in a DOS-box (protected-mode) or in real-mode MS-DOS which causes a complete unloading of Win95 and re-loading after finishing the DOS-application. It also supports a boot menu for booting real-mode DOS directly.

ERTS will only be accurate in real-mode MS-DOS but will also run in protected-mode MS-DOS within a full-screen (!) DOS-box. In the later case, timing and screen sync will be accurate enough for testing experiment definitions.

9.3.1 Activating ERTS from within Win95

Create a short-cut and edit properties (actually editing PIF files). Set Full-screen mode, deactivate any Win95 key assignments, set IDL-time to low.

Real-mode DOS (accurate)

If you want to start ERTS in real-mode DOS (accurate), set advanced program options to MS-DOS. In this case you even have the opportunity to define special CONFIG.SYS and AUTOEXEC.BAT files (e.g. loading DOS SoundBlaster drivers).

Protected-mode DOS (not accurate, but comfortable)

If you want to start ERTS protected-mode DOS (not accurate, but fast turn-around), then set the advanced program options to "Mode suggested by Win95" (default).

9.3.2 Activating ERTS from the DOS prompt.

Selecting prompt mode from the boot menu

(1) edit the file msdos.sys in your boot directory by first removing the „shr“ flags with ATTRIB and then editing the file with EDIT. Insert BootMenu=1 into the [option] section. This will activate the Win95 boot menu each time you start the computer. You can select normal mode or prompt mode at this point. (same as pressing F8 during boot phase).

Restarting the computer in prompt mode

(2) Select "Shut down" from in the Start menu of Win95 and select restart in prompt mode (pressing SHIFT key while clicking on OK speeds restart up)

Creating a short cut for MS-DOS in real mode

(3) Create a second short cut in the start menu by copying the default MS-DOS short cut in the start menu and rename this to e.g. "Real MS-DOS". Then edit the advanced program options and set execution mode to MS-DOS.

10 New to ERTS V3.23

10.1 General ERTS commands

10.1.1 User-defined FB-classes for DROPOUT mode

The definition of drop-out modes for blocks has been extended to support user-defined classification of correct/false responses with respect to the twelve pre-defined FB-classes.

Syntax: DROPOUT [ValMode [TstMode [ValRep [TstRep [FB1 ... FB12]]]]]

Mode Determines the drop-out mode for valid trials (Val) and WarmUp trials (Tst)

- 0 No drop-out mode (default)
- 1 One drop-out run without storing incorrect trials
- n > 1 Dropout run with storing all trials
Max number of trials is n x N(Trials)

Rep Determines how incorrect trials are repeated for valid (Val) and WarmUp trials (Tst)

- 0 repeat at end of block (default)
- 1 repeat immediately

FB? Determines which FB-class should be dropped

- 0 Repeat (default for all FB-classes except 2)
- 1 Drop (default for FB-class 2)

The separation of drop-out mode and randomization mode allows the user to combine all drop-out modes with all randomization modes, which was not possible before that. The block randomization modes 4 and 5 will be not longer supported.

10.1.2 New cancel mode for nested trials

The cancel flag of the RT (Run Trial) trial command has been extended to support abortion of nested trials at the end of a trial.

Syntax: RT Trial [Loops [Cancel [Voice [RndLoop [VarArgList]]]]]

- Cancel 0 Do not abort
- 1 Immediate response dependent abortion
- 2 Response dependent abortion at end of trial

10.1.3 Aborting block with STOPKEY session command

The STOPKEY session command now supports not only PC keyboard but all input devices (except voicekey). Besides interrupting a running session this command may now also be used to abort a running block.

Syntax: STOPKEY [StopKey [Trial [VarArgList]]]

Trial Name of trial definition that is executed to interrupt the session
If null '-' then no trial will be executed and the running block will be immediately aborted even within a trial. Trials up to this moment are recorded.

10.1.4 New behavior of MAXTIME

If the MAXTIME session command is used to abort blocks then the block will be immediately aborted within the running trial. Previously, MAXTIME waited until the end of the running trial. The aborted trial is not recorded.

10.1.5 New ET (End Trial) trial command

To abort a running trial depending on the response of the subject, the new ET trial command has been implemented. The command can only be used as a FB alternative within a FB command or within a trial called by a FB command.

Syntax: ET [Mode]

Mode 1 Abort trial
2 Abort trial and repeat immediately

10.1.6 Support of VESA-VBE V2.0 video modes in ERTSVIPL

To support other resolutions than the predefined ERTS video modes, the selection of a video mode may now be given based on the resolution instead of the video number. The syntax of the command line option /H has been extended.

Syntax: /Hn or /H[w]x[h]

n Number of video mode

- 1 320 x 200
- 2 640 x 400
- 3 640 x 480
- 4 800 x 600
- 5 1024 x 768
- 6 1280 x 1024

w Width in pixel of desired resolution

h Height in pixel of desired resolution

Note that if one dimension is unambiguous then it is sufficient to describe only this value. You must, however, include in any case the 'x' to tell ERTS that this is a resolution and not a video mode.

10.1.7 New support of sine wave tones in units of Hertz

A new type of tone definition supports the presentation of sine wave tones in units of Hertz via SoundBlaster compatible soundcards (OPL-chip required). There are three independent sources of sine waves: Mono, Stereo Left, and Stereo Right.

For varying source, volume and frequency independently of each other, there are three subtypes of Hertz tone definitions.

Syntax: TONE name H[subtype]

subtype	t	Tone definition (default)
	v	Volume definition
	s	Source definition

The volume and source subtypes only set the characteristics for subsequent tone presentations similar to mixer control definitions. Only the tone definition results in the presentation of a sine wave tone.

Tone definition

This type is the default type and results in the presentation of the sine wave tone when used with the ST-trial command

Syntax : LeftPitch [RightPitch]

LeftPitch	Frequency in Hertz of left source or mono source
RightPitch	Frequency in Hertz of right source

Because the FM chip of the SoundBlaster card cannot programmed to any frequency there is a new command line option for displaying the frequency that is realized by the FM chip.

Syntax: /Ih (Info about Hertz)

Volume definition

This type sets the attenuation of each source in units of -.75 dB. A Null means maximum volume and a 63 means 47.25 dB attenuation.

Syntax: LeftAttenuation RightAttenuation

LeftAttenuation	0-63, volume of left source or mono source in units of -.75dB, default is 0 (max volume).
RightAttenuation	0-63 volume of right source in units of -.75dB, default is left attenuation

Source definition

This type selects the source of sine wave tones.

Syntax: Source

0	Mono
1	Right stereo source
2	Left Stereo source
3	Both stereo sources

How left/right frequency and volume settings will be interpreted depends on the selected source. If mono is selected, only the LeftAttenuation and LeftPitch arguments are used to define the mono tone. If only one of the stereo channels is selected, only the corresponding arguments will be considered. Stereo sources require two OPL2 chips or one OPL3 chip.

10.1.8 Frequency based VoiceKey

The ERTS voicekey has been enhanced to support frequency based thresholds as well. A zero-crossing threshold may be specified by two parameters, MaxCross the number of zero-crossings and MinAmp, the smallest amplitude size that a sound sample must exceed to be counted as zero-crossing.

A zero-crossing algorithm detects high-frequency noise even at volume levels below the volume threshold, for example, the early detection of spoken words like „Five“, „Seven“ will be improved. The theoretical resolution of the ERTS on-line zero-crossing algorithm is about 5 ms.

The VoiceKey parameters may be permanently specified within the ERTS configuration file, or by using the new VOICEKEY system command within an ERTS script.

Syntax: VOICEKEY MaxVol [MaxCross [MinVol]

MaxVol	Volume threshold (0-127, default 25). A null turns off volume-based threshold
MaxCross	Frequency threshold (0-65535, default 25). A null turns off frequency-based threshold A value of 30-50 is a reasonable value (see TESTVKEY utility).
MinVol	Minimum volume threshold for counting zero-crossings (0-127, default 2)

The VOICEKEY command overrides settings stored in the ERTS.CFG file (see ERTSCFG configuration utility).

10.1.9 Extended FONT command to reduce memory usage

The FONT command has been extended by a optional argument to list used characters. If this argument is appended on the font definition ERTS will only load bitmaps for characters included in this list. Use this new argument to reduce the amount of conventional memory needed to run a script.

The blank character (dec 32) is always loaded.

Syntax: FONT Name
FontFile[.Ext] Size [Points [Comp [LineSpace [CharSpace [UsedChars]]]]]

UsedChars	This argument lists character which will be used with this font. No other character bitmaps are loaded except the blank character (dec 32). The list of used characters must not include a blank.
-----------	---

10.1.10 Modified FB-command

The meaning of Sequential argument (new name CompMode) has been slightly changed. If you have used an argument setting larger than 1 to reduce the comparison of keys to one particular input, you must rewrite the script by using the new argument KeyIndex of the FB-command:

FB CorrKey [CorrNo [CompMode [RtMin [RtMax [RtMode [KeyIndex]]]]]]

CompMode	Determines how keys are compared in multiple key input situations
0	Correct is if one of the reponses matches one of the correct keys
1	Correct is if the order of input corresponds to order of correct keys
2	Correct if all correct keys appear in key input independent of order.
KeyIndex	Determines which latency will be compared in multiple key input situations
0	Compare all input
i > 0	Compare only i-th input

10.1.11 User-defined index for background color

The SCREENCOLOR system command has been extended to support a user-defined index for the ERTS-background color.

SCREENCOLOR Fore [Back [Index [PicPalette]]]

Fore	Foreground color.
Back	Background color
Index	Palette index (0-255) for ERTS background .
PicPalette	Name of BMP or RLE graphics file from where ERTS should extract the background color according to 'Index'.

The new arguments are helpful when working with 256-color pictures because those pictures overwrite the entire ERTS color palette with their color palette. In order to match the ERTS-background with the background of the pictures you must set the same background index and color in ERTS as it is used in the pictures. By default ERTS uses palette index 0 for the background color when 256-color are displayed.

To match the background color (not the index) you may either define a corresponding color definition or name a BMP/RLE graphics file in the optional 'PicPalette' argument'. In this case, ERTS reads the background color out of the picture color palette.

10.1.12 User-defined index for transparent color

When displaying color graphics pictures defined as transparent (type Gt) ERTS normally loads the first color entry of the color palette (index 0) as transparent. If the background color of pictures is located on another offset within the color palette you may now add a new optional palette index to the picture definition.

PICTURE name Gt[index] datei

Index	Palette index of transparent color (0-255, default 0)
-------	---

10.2 ERTS and MS-Windows 95

Win95 provides much better support for activating DOS-applications from within Windows than Win 3.x. You can start DOS-applications in a DOS-box (protected-mode) or in real-mode MS-DOS which causes a complete unloading of Win95 and re-loading after finishing the DOS-application. It also supports a boot menu for booting real-mode DOS directly.

ERTS will only be accurate in real-mode MS-DOS but will also run in protected-mode MS-DOS within a full-screen (!) DOS-box. In the later case, timing and screen sync will be accurate enough for testing experiment definitions.

10.2.1 Activating ERTS from within Win95

Create a short-cut and edit properties (actually editing PIF files). Set Full-screen mode, deactivate any Win95 key assignments, set IDL-time to low.

Real-mode DOS (accurate)

If you want to start ERTS in real-mode DOS (accurate), set advanced program options to MS-DOS. In this case you even have the opportunity to define special CONFIG.SYS and AUTOEXEC.BAT files (e.g. loading DOS SoundBlaster drivers).

Protected-mode DOS (not accurate, but comfortable)

If you want to start ERTS protected-mode DOS (not accurate, but fast turn-around), then set the advanced program options to "Mode suggested by Win95" (default).

10.2.2 Activating ERTS from the DOS prompt.

Selecting prompt mode from the boot menu

(1) edit the file msdos.sys in your boot directory by first removing the „shr“ flags with ATTRIB and then editing the file with EDIT. Insert BootMenu=1 into the [option] section. This will activate the Win95 boot menu each time you start the computer. You can select normal mode or prompt mode at this point. (same as pressing F8 during boot phase).

Restarting the computer in prompt mode

(2) Select "Shut down" from in the Start menu of Win95 and select restart in prompt mode (pressing SHIFT key while clicking on OK speeds restart up)

Creating a short cut for MS-DOS in real mode

(3) Create a second short cut in the start menu by copying the default MS-DOS short cut in the start menu and rename this to e.g. "Real MS-DOS". Then edit the advanced program options and set execution mode to MS-DOS.

10.3 TESTVKEY V1.03 for testing voice-key thresholds

For easier adjustment of the voice-key threshold and the SoundBlaster input mixer settings, the new TESTVKEY utility has been designed. TESTVKEY visualizes the current input volume and may be called with different threshold values for optimizing voice-key sensitivity.

Usage: TESTVKEY MaxVol [MaxCross [MinVol]]

MaxVol	Volume threshold (0-127, default 25). A null turns off volume-based threshold
MaxCross	Frequency threshold (0-65535, default 25). A null turns off frequency-based threshold A value of 30-50 is a reasonable value (see TESTVKEY utility).
MinVol	Minimum volume threshold for counting zero-crossings (0-127, default 2)

The frequency threshold is based on a zero-crossing algorithm which counts pos/neg amplitude peaks larger than MinVol. The number of zero-crossings is also displayed (Cross). The frequency threshold is very sensitive to high pitch noise even at low volume levels (e.g. spoken word „five“).

Use volume threshold and frequency threshold together to optimize voice-key performance. MaxVol should be set to a value close to random noise level but sufficiently high to avoid false alarms. MaxCross should be set to a value close to random zero crossings but sufficiently high to avoid false alarms. Note that number of zero crossings depend also on MinVol. MinVol should be set to about the random volume level or even lower to make the frequency based threshold as sensitive as possible. If MinVol is too high, no zero-crossings will be observed at low volume.

Try different thresholds and mixer settings to find the best configuration for a sensitive voice-key without false alarms. Configure ERTS with the best threshold values using the ERTSCFG utility. Be sure to set mixer settings prior to ERTS sessions.

11 ERTS V3.18

11.1 General ERTS commands

11.1.1 New result header format

The header of ERTS result files has been slightly changed. It now contains an info line about the monitor characteristics like resolution, size, and refresh rate. The “version“ line has been removed moving all following entries one line upwards. Overall number of header lines remained identical except when random seed value is logged.

11.1.2 SE-trial command (SET ECHO)

Description of the PosNo and Case argument of the SE-trial command in the Reference manual is not complete. See SETECHO session command for a complete description of argument PosNo and Case.

There is also a new argument ‘Prompt’ for both commands:

Syntax: SETECHO Position [PosNo [Font [Case [Prompt]]]]

Position	see manual
PosNo	Index of position within the position definition. Default 1. If 0, i-th keystroke will be displayed on i-th position index.
Case	Determines case of echo <ol style="list-style-type: none"> 0 Lower case 1 Upper Case (default) 2 Mixed Case
Prompt	Determines the display of empty echo field as underlined input field <ol style="list-style-type: none"> 0 Do not display empty echo field 1 Display empty echo field with underscore characters (default)

11.1.3 New SW-trial command (Single Word presentation)

The new SW trial command provides the functionality of presenting text pictures in a word-wise mode. The SW-command calls a user-defined trial definition for each blank-separated token of the text picture. Thus, events for each single word presentation can be defined in a very flexible way by just creating the appropriate trial definition.

Syntax: SW TextPic MaxWords SWmode SWMask SWTrial [VarArgList]

TextPic	Name of text picture that should be presented in single word mode. Trial commands within SWTrial, which display TextPic, automatically refer to the single word.
MaxWords	Maximum number of words in one text picture. Can be set to 0 if trial does not contain response registration
SWmode	Determines the location of the single words <ol style="list-style-type: none"> 1 Words are displayed superimposed on the same location 2 Words on original location within sentence. If words are presented in OR-mode, entire sentence will be build word-wise 3 Increment sentence word-wise (similar to mode 2 in OR-mode). Use this mode if mode 2 in combination with OR-mode is not suitable (e.g. if masking is turned on).

- SWmask** Name of a picture set which contains one or two pictures as elements. These pictures are used for cuing or masking word locations. The first picture within the set definition will be used for forthcoming sentence positions, The second picture, if defined for past words. To disable masking completely use a hyphen (-).
- SWTrial** Name of trial that should be executed for each word. Corresponds to the body of the loop which cycles through one sentence. Details like fixed-rate or self-paced presentation must be defined in this trial. The SP command that presents TextPic splits automatically the contents into word tokens.
If a WV command is used within this trial, then ERTS will automatically overwrite the marker value with the single word loop counter. If a VOC file with markers behind each word is started prior to the SW command, then single word presentation can be timed by marker values. Markers must begin with 1 and increment by 1.
- VarArgList** Values for variable trial arguments. Typically the TextPic will be one variable argument

Example:

```

PICTURE FS T
This sentence is presented at a fixed rate
with superimposed words

POSITION Center P 0 0

; sample trial definition for fixed presentation

TRIAL Fixed ; This trial will present single words at a fixed rate
SP ? Center 1 ; This SP will present one single word if called from SW
ST 500 100 0 ; A tone with each word just for fun
LB ; Loading border to avoid preloading between SP and ST
CT ; Clear tone after 100 ms
+ CS 200 ; Show Pic for another 100 ms

TRIAL ShowFixed
SF Nice
CS 1000
SW FS 0 1 Fixed FS ; IMPORTANT: Name of trial must not be variable
; if trial requires a VarArgList

```

11.1.4 Support of additional screen images with standard ERTS

Like ERTSVIPL, standard ERTS now also supports the trial commands LI, SI, and TI to load and display selected full screen images in addition to the automatic IPL double buffers. Consequently, you may use additional images in the same way for realizing tachistoscopic displays with both versions of ERTS. Color manipulation technique is not necessary anymore (see section 7.5 of the User's manual).

Unlike ERTSVIPL, standard ERTS offers always up to two additional screen images independent of available video RAM. While screen images are loaded, multiple foreground colors cannot be realized. Thus, clear (CS) unused screen images if you want to display multiple color screens after using the additional image.

11.1.5 Enhanced color palette updating features for ERTSVIPL

For updating the 256-color palette without writing a new picture bitmap, the following features have been implemented: new CP-trial command (Change Palette), new picture subtype „c“, and enhanced CC-trial command for overwriting any of the 256 color-palette entries directly.

New CP-trial command

Syntax: CP Picture [Time [Cancel [Offset]]]

Default	Time	0
	Cancel	0
	Offset	0

Picture	Name of BMP or RLE graphics picture whose color palette should be used to update ERTS system color palette. Picture may be of subtype „c“.
Time	Time that will be waited after updating palette.
Cancel	Cancel mode
Offset	Offset (0-255) within system palette.

When updating 16-color bitmaps you must find the correct offset within the dynamic color palette. 16 color palettes are assigned to the internal system palette in units of 16-entries. ERTS loads multiple 16-color bitmaps in ascending order beginning with offset 0, if no 256-color bitmap is used somewhere in the experiment, or beginning with offset 16, if 256-color images are used. Thus, the 16-color palette which you want to update might be located at offset 0, 16, 32, and so on depending on the order it was loaded for preparing this particular screen image. If color palette of picture bitmaps are identical, they will share the same system palette entries.

New picture subtype „C“:

For defining a color palette only without picture bitmap you may use the new subtype „c“ in combination with the „G“ picture type and the BMP or RLE file format. ERTS will only load the color table but not the bitmap. You, therefore, may use this picture definition only for the CP-command.

Enhanced CC-trial command

Syntax: CC ChangeMode [Hue [Saturation [Value [Level [DestColor]]]]]

The valid range of values for the level command has been extended:

Level	
0	Background
n > 1	n-th loaded foreground color (max 3 for standard ERTS)
n > 99	n-th - 100 color palette entry (ERTSVIPL only)

Whereas level 1 to n < 99 is intended to address foreground colors loaded internally with the color argument of visual trial commands, levels 100 to 355 are intended to address the color palette entries 0-255 explicitly.

11.1.6 Revised voice-key implementation

The implementation of the SoundBlaster voice-key has been slightly revised because some user's reported problems activating the voice-key. To make the voice-key more stable, the size of the internal transfer buffer has been increased and the threshold must be crossed by at least two values to avoid false alarms by single faulty byte values which have been observed with some mixer settings.

Resolution and accuracy of voice-key registration is 1ms. The resolution for triggering trial event timing and for key coding is, however, only 200ms. Thus, if you are waiting for a voice-key response within a trial sequence, the following event might be late by as much as 200 ms after threshold was actually crossed. This is due to the internal buffering of incoming sound data.

11.1.7 New KEYCODE session command

The new KEYCODE session command may be used to assign parallel port bit patterns and/or acoustic signals to keystrokes.

Syntax: KEYCODE KeySet [ToneSet [ByteSequ [Channel [Bios]]]]

KeySet	Name of a key definition which contains keys to be coded.
ToneSet	Name of a set which contains tone definitions for each of the keys. (default none)

ByteSequ Name of a byte sequence which contains bytes for each of the keys. (default none)
 Channel Channel number 1-4 (translated by ERTSCFG to parallel port numbers). Default 1
 Bios Determines handshake (see SS command) (default 0)

After activating key coding ERTS will send the i-th byte from ByteSequ to the parallel port if the i-th key within KeySet is pressed. In the same way, the i-th tone definition within ToneSet will be played. If ToneSet or ByteSequ contain less members than KeySet, the largest available entry will be selected. Thus, if you want to code all keys with the same byte, you may define a ByteSequ containing only this single byte

Note that key coding happens independent from the trial events, which means that during the entire session keys will be coded until another KEYCODE command disable key coding or until end of session. To code the release of a key, you must simply include the release code for that key in KeySet. For setting single bits it is important to use masked byte definitions (see SEQUENCE data command).

```
SEQUENCE CodeSequ B
nnnnnn1n nnnnnn0n nnnnnnn1 nnnnnnn0

KEY CodeKeys
SHL ^SHL SHR ^SHR

SET CodeTones T
400 800 500 900

PICTURE Instruct T
Press SHL or SHR to test key coding

Press SPACE to Exit

POSITION Center P 0,0

TRIAL Test
SP Instruct Center
WK Space

SESSION
KEYCODE CodeKeys CodeTones CodeSequ 1 0
RUNTRIAL Test
```

11.1.8 New WP-trial command (Wait for parallel port)

The new WP-trial command allows synchronization of trial events with external trigger signals which are read from the status lines of one of the parallel ports. Note that these trigger signals cannot be used for response registration but only for synchronization. Use the EXKEY-logic for registering external TTL-signals as responses.

Syntax: WP BitPattern [Channel [Cancel]]

BitPattern Name of a sequence definition which contains exactly one masked byte.
 Channel Channel number 1-4 (translated by ERTSCFG to parallel port numbers). Default 1
 Cancel Cancel mode (see other trial commands for description). Default 0

The WP-command waits until the state of the status lines matches the specified bits. Masked bits are ignored. You may use up to five different status lines:

Line	Bit	Value	Pin at Connector
BUSY	4	16	11
ACK	3	8	10
PE	2	4	12
Select	1	2	13
Error	0	1	15

The following example illustrates how to use the WP command to wait for a signal at the BUSY line.


```

SEQUENCE IsBusy B
nnn1nnnn

SEQUENCE IsReady B
nnn0nnnn

PICTURE WaitForBusy N
PICTURE WaitForReady N

POSITION Center P 0,0

TRIAL TestWP
SP WaitForBusy Center
WP IsBusy
SP WaitForReady
WP IsReady

```

11.1.9 Extended /I command line option

The syntax and meaning of the /I option has been changed. The option is now meant to be a general info option. Just as with the /O option, a second argument specifies the type of information requested.

```

Is Information about session structure (previous /I)
It Information about internal trial structure (previous /Tt)
If Information about font loading (previously included in /W)
Im Information about free memory
Ix Information about extended memory allocation

```

11.1.10 Reduced program memory requirements

The memory required by ERTS has been greatly reduced (about 70KB) by implementing a program overlay structure. The overlay structure has been designed to avoid disk access during trial and block execution. When loading experiment files you can reduce loading/reloading overlays by grouping similar commands together within the experiment file (e.g. all pictures, all positions, all trials, all blocks, session).

A side effect of the overlay manager is the allocation of an overlay cache in the extended memory. If this causes memory problems, you may reduce or disable the cache by defining the DOS-environment entry

```
MOVE_XMS=xxxx
```

where xxxx is a four digit hexadecimal number > 0000 specifying the size of the cache in KB. To disable the cache use 0001.

11.1.11 New DROPOUT session command

The definition of drop-out modes for blocks has been separated from the RUNBLOCK session command. Drop-Out modes are not longer one variant of the randomization modes but can be defined independent of the randomisation mode with the new DROPOUT session command. This command enables or disables drop-out mode for all subsequent RUNBLOCK commands.

Syntax: DROPOUT [ValMode [TstMode [ValRep [TstRep]]]]

Mode Determines the drop-out mode for valid trials (Val) and WarmUp trials (Tst)

```

0 No drop-out mode (default)
1 One drop-out run without storing incorrect trials
n > 1 Dropout run with storing all trials
Max number of trials is n x N(Trials)

```

Rep Determines how incorrect trials are repeated for valid (Val) and WarmUp trials (Tst)

```

0 repeat at end of block (default)

```

- 1 repeat immediately

The separation of drop-out mode and randomization mode allows the user to combine all drop-out modes with all randomization modes, which was not possible before that. The block randomization modes 4 and 5 will be not longer supported.

11.1.12 New Hercules session monitoring

An online session monitoring (HERCLOG) is now supported with dual monitor hardware configurations. The experimentator can permanently watch the status of the session on the monochrome monitor. The session monitoring is activated by the following new command line option

/M[n] Activates hercules session monitoring

- | | |
|---|--|
| n | |
| 1 | Erase HERCLOG at end of session and return to DOS |
| 2 | Blink and wait for key press at end of session. First key press stops blinking, second key press returns to DOS. |

11.1.13 Slightly changed meanings of statistics within feedback picture

The Pt statistic (Percent(total)) now represents the percentage of executed trials in relation to the total number of trials in the session.

If a trial does not contain any FB-command, latencies will be assigned to totals but neither to correct nor to false.

11.1.14 Extended FB-command

The meaning of the Sequential and Begin/End argument has been extended:

Sequential Determines how input is compared to correct keys in multiple key input situations

- | | |
|-------|--|
| 0 | Correct is true if one of the reponses matches one of the correct keys |
| 1 | Correct is true if the order of input corresponds to order of correct keys |
| i > 1 | Compare only i-th input with correct keys |

Begin/End Determines which latency will be compared in multiple key input situations

- | | |
|---|--|
| 0 | Use first registered input |
| 1 | Use last registered input |
| 2 | Use difference between last and first registered input |

11.1.15 New optional prompts

The default flag of the PROMPT system command has been extended to support optional input in response to prompts:

- | | |
|---------|--|
| default | Determines default mode of prompt |
| 0 | No default value |
| 1 | Suggest default value |
| 2 | Use default value without prompting (used in combination with PARAMETER command) |

- 1 No default value, prompt is optional and may be answered just by pressing RETURN key.

11.1.16 New optional session comment

A new Type of the PROMPT system command has been created to support input of an optional session comment which, if defined, is added to the result table header

type	Determines type of input data (case sensitive)
C	Text input which is used as comment

see manual for other types

11.2 ERTSCODE

11.2.1 New pre-defined text constant

The new HEADCOMMENT text constant holds the content of the new optional ERTS comment line if present in the result table header.

11.3 New utility TESTVKEY for testing voice-key

For easier adjustment of the voice-key threshold and the SoundBlaster input mixer settings, the new TESTVKEY utility has been designed. TESTVKEY visualizes the current input volume and may be called with different threshold values for optimizing voice-key sensitivity.

Usage: TESTVKEY threshold

threshold value between 1 and 127

Use TESTVKEY in combination with the SoundBlaster utility SB16SET to try different combinations of input gain, automatic gain control (AGC), and voice-key threshold.

11.4 New utility TESTTIME for testing millicsecond timer

The new TESTTIME utility has been designed for testing accuracy and reliability of the ERTS millisecond timer in different operating systems and hardware configurations. The main purpose is to detect distortions caused by multitasking operating systems, time consuming background processes, or defective timer chips.

Usage: TESTTIME [tolerance] [Nerrors] [SyncFlag]

tolerance	Largest time period between two successive measurements which are tolerated. (default 1ms). (Must be set to screen refresh rate if SyncFlag is active)
Nerrors	Number of out-of-range errors until TESTTIME aborts execution
SyncFlag	Synchronize each measurement with vertical retrace of the screen.

TESTTIME continuously reads the millisecond timer and checks the time period between two measurements. If this period is beyond or below the tolerance range, the error count is incremented. After Nerrors the test run will be aborted. An interval between two measurements is typically several microseconds. Some small variability due to memory refresh and hardware interrupts is normal and should not exceed one millisecond. Set the SyncFlag to test screen sync and accuracy in combination with a slower measurement rate.

If timing is correct, TESTTIME should run several minutes without aborting. Recommended length for one test run is between 30 and 60 seconds. If any distortions exist they usually show up immediately after several

seconds. TESTTIME aborts and displays the last 100 measurements, the internal timer tic values, and a frequency distribution of the measurements. Note that the scale of the frequency table is dynamically extended which implies the loss of previous samples.

11.5 New utility MERGEVOC for dichotic tasks

MERGEVOC is a utility for generating sound samples for dichotic tasks prior to the experimental session. Two mono VOC-files are mixed to left/right stereo channels with a user-defined delay of one channel. MERGEVOC can be called.

MERGEVOC supports all sampling rates and uncompressed 8/16-bit format. Sound samples must consist of one data block only. Both mono input files must have the same format with respect to sampling rate and bits per sample. MERGEVOC can be called batch-like in quiet mode to generate sound samples for an experimental session. If number of combinations exceeds disk space experimental sessions must be split into subsets.

Same functionality will be implemented in ERTS in one of the next releases by a new trial command that merges two mono sound files into one output stereo file.

Syntax: MERGEVOC left.voc right.voc out.voc SOA [-q]

Left.voc	Name of VOC-file for left channel
Right.voc	Name of VOC-file for right channel
Out.voc	Name of output VOC-file
SOA	Delay between channels in ms (negative means left first)
-q	Optional option for suppressing system messages

11.6 ERTSLIC Serialization

The new version of ERTSLIC now supports deinstallation as well:

```
ERTSLIC [/Remove]
```

```
    /Remove    Current license will be removed from the machine
```

When removing a license ERTSLIC will generate a deinstallation number that must be reported to your local distributor for confirming deinstallation.

ATTENTION: New serializations require new serial numbers that will be supplied by your local distributor only! Deinstallation number will only be generated on machines that have been serialized with this new version of ERTSLIC.