manual
for
eprimer version 0.5

Hester Breman and Tobias Otto

November 8, 2011

Contents

1 Introduction and installation

This is a manual for using `eprimer`, a BrainVoyager QX GUI plugin, to convert an E-Prime Log file (*.txt) to a BrainVoyager QX stimulation protocol (*.prt). This will be available at the following BrainVoyager support page. For more information on the BrainVoyager QX stimulation protocol, see step 3 of the Getting Started Guide that can be downloaded via the 'Downloads' section of the BrainVoyager website. E-Prime is software to create and run an experiment for computer or MRI scanner. For more information, please consult the Psychology Software Tools website.

1.1 Installation

Installation can be performed by extracting the *.zip file and placing the *.dylib/dll, *.ui and *.js files in
`/Documents/BVQXExtensions/Plugins_32,64/`.

After installation, `eprimer` can be started via the BrainVoyager "Plugins" website (see figure 1).

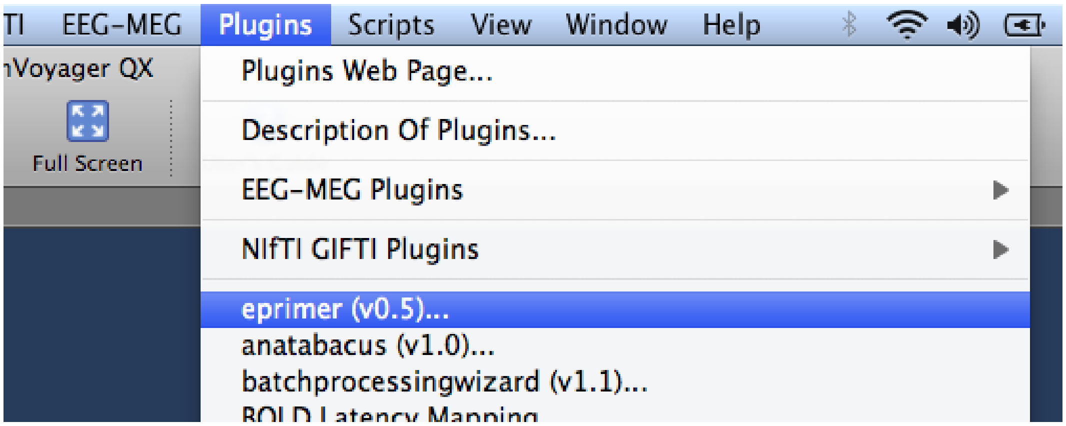


Figure 1: After installation, the tool can be started via the BrainVoyager "Plugins" menu

1.2 Notes on the current version

v0.5: This version is updated to work with the new scripting system in BrainVoyager QX 2.2 and higher. The configuration of conditions can now be performed just via a configuration text file.
v0.4: The configuration of the conditions is now automatically saved after configuring the conditions via a dialog. This configuration can then be re-used for the next subjects in the same experiment.
v0.2: One can define conditions independent of the logged variables. Because of this possibility, the end time of an interval should be defined with respect to the start time of the first pulse, for example using the property "FinishTime"; this means that selecting "RT" as end of an interval will not deliver valid output (since "RT" has a timing parameter that is relative to the same variable).
v0.1: The protocol is created with milliseconds resolution.

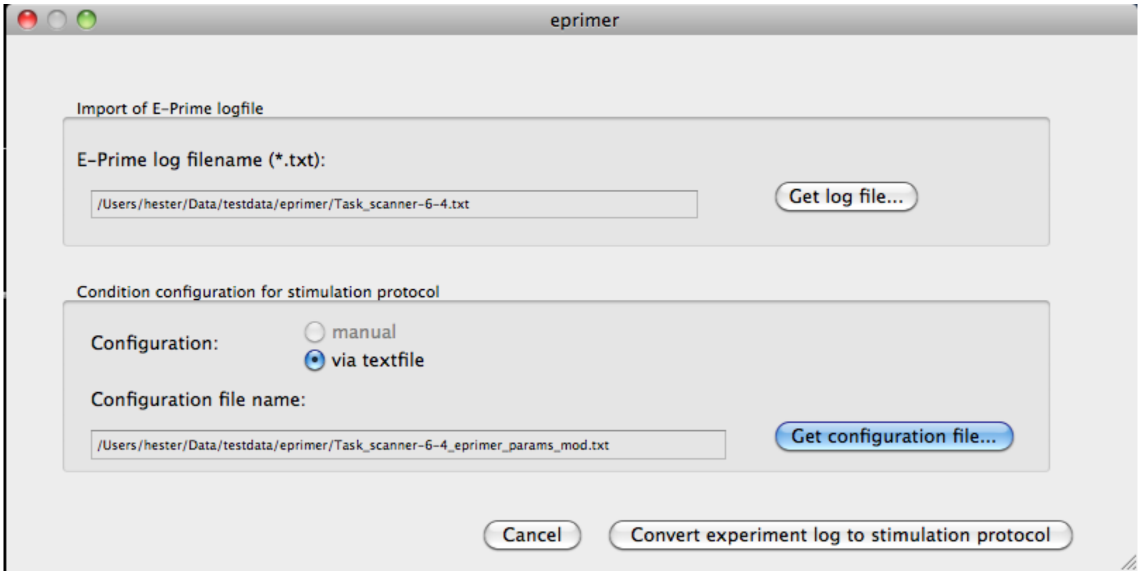
1.3 Acknowledgements

This tool was developed in collaboration with Tobias Otto (University of Maastricht, The Netherlands).

2 Usage

2.1 Start

Start `eprimer` via the BrainVoyager QX "Plugins" menu. A dialog will appear with the buttons for loading the two text files that are required for the conversion from E-Prime Log files (*.txt) to BrainVoyager stimulation protocols (*.prt).


eprimer
Import of E-Prime logfile
Get log file...
Condition configuration for stimulation protocol
Get configuration file...
Cancel
Convert experiment log to stimulation protocol

E-Prime log filename (*.txt): /Users/hester/Data/testdata/eprimer/Task_scanner-6-4.txt

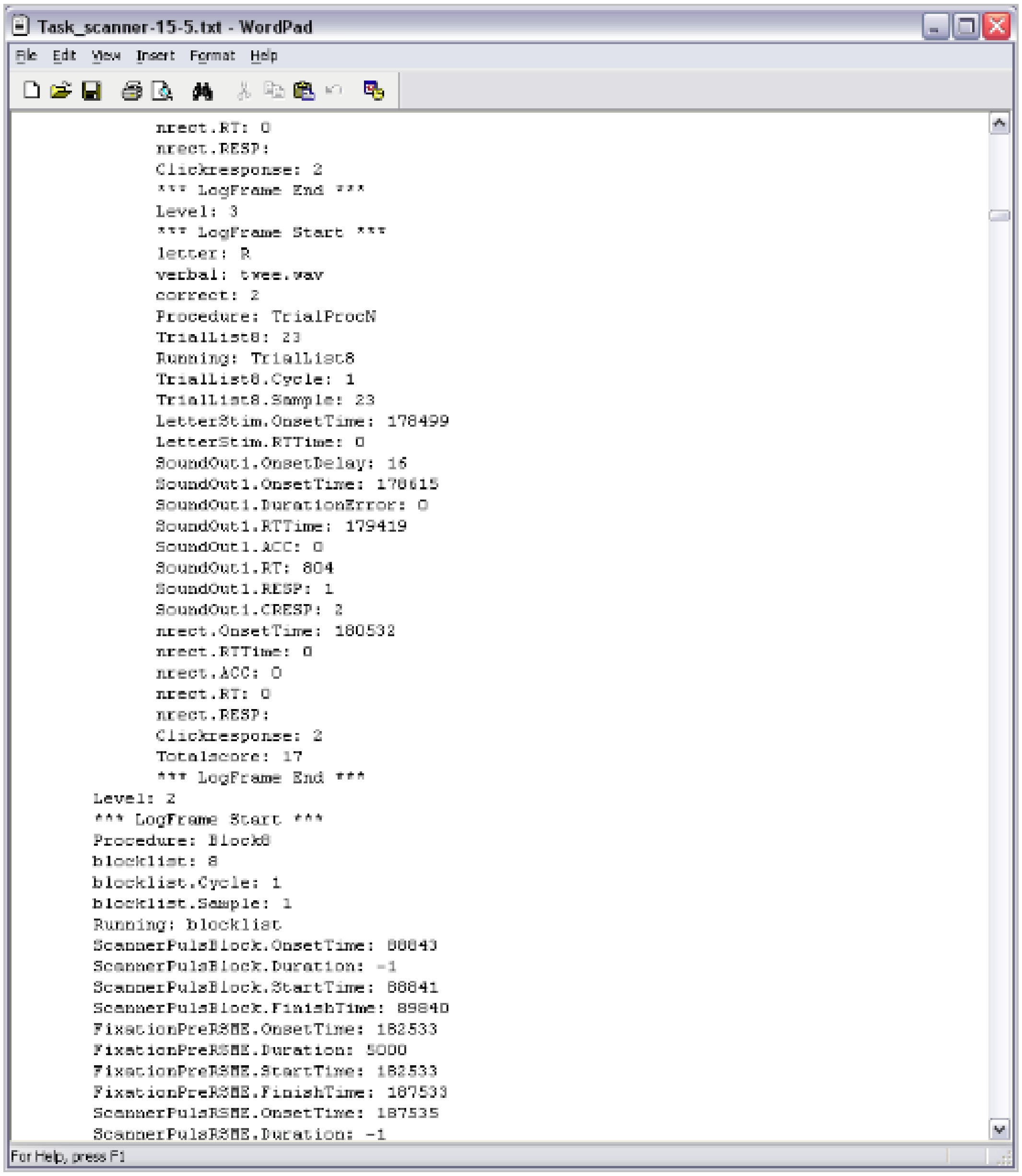
Configuration: ○ manual ● via textfile

Configuration file name: /Users/hester/Data/testdata/eprimer/Task_scanner-6-4_eprimer_params_mod.txt

Figure 2: Main menu of tool

2.2 Load the E-Prime Log file

One can select the E-Prime Log file (*.txt, for an example see figure 3) via the first button. After selection of the file, the name will become visible in the first box.



```
      nrect.RT: 0
      nrect.RESP:
      Clickresponse: 2
      *** LogFrame End ***
      Level: 3
      *** LogFrame Start ***
      letter: R
      verbal: twee.wav
      correct: 2
      Procedure: TrialProcN
      Triallist8: 23
      Running: Triallist8
      Triallist8.Cycle: 1
      Triallist8.Sample: 23
      LetterStim.OnsetTime: 178499
      LetterStim.RTTime: 0
      SoundOut1.OnsetDelay: 16
      SoundOut1.OnsetTime: 178615
      SoundOut1.DurationError: 0
      SoundOut1.RTTime: 179419
      SoundOut1.ACC: 0
      SoundOut1.RT: 804
      SoundOut1.RESP: 1
      SoundOut1.CRESP: 2
      nrect.OnsetTime: 180532
      nrect.RTTime: 0
      nrect.ACC: 0
      nrect.RT: 0
      nrect.RESP:
      Clickresponse: 2
      Totalscore: 17
      *** LogFrame End ***
Level: 2
*** LogFrame Start ***
Procedure: Block8
blocklist: 8
blocklist.Cycle: 1
blocklist.Sample: 1
Running: blocklist
ScannerPulsBlock.OnsetTime: 88843
ScannerPulsBlock.Duration: -1
ScannerPulsBlock.StartTime: 88841
ScannerPulsBlock.FinishTime: 89840
FixationPreRSME.OnsetTime: 182533
FixationPreRSME.Duration: 5000
FixationPreRSME.StartTime: 182533
FixationPreRSME.FinishTime: 187533
ScannerPulsRSME.OnsetTime: 187535
ScannerPulsRSME.Duration: -1
```

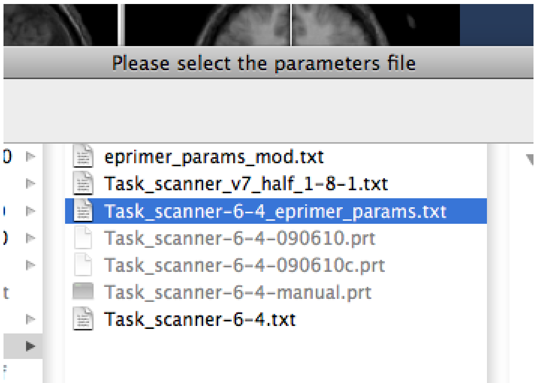
Figure 3: E-Prime log file

2.3 Configuring conditions

The names and properties in an E-Prime log file that will define a condition can be fed to `eprimer` via a plain text file (*.txt).

2.3.1 Configuring conditions via a textfile

To configure the conditions according to the parameter text file, the lower button on the dialog can be used to load the parameter file. After the file has been selected via a file dialog (see figure 4), the name of the file will become visible in the lower box on the dialog.



Please select the parameters file

eprimer_params_mod.txt
Task_scanner_v7_half_1-8-1.txt
Task_scanner-6-4_eprimer_params.txt
Task_scanner-6-4-090610.prt
Task_scanner-6-4-090610c.prt
Task_scanner-6-4-manual.prt
Task_scanner-6-4.txt

Figure 4: File selection dialog for eprimer example parameter file

2.3.2 Format of the configuration textfile

The text file starts with a header of exactly 5 lines. This header can contain any type of information one would like to write here (or it can be kept empty). After these 5 lines, the names of the parameters are printed, with on the next line(s) the values for the parameters. All parameters should have the same number of values as number of conditions, except for the variables `number of conditions` which indicates the number of conditions, `experimentname` which is a single name, and `zero` which is a single number, indicating the start time of the experiment (see appendix B). Also, the order of the properties for each condition should be similar for each type of parameter (color for first condition coming first, color for fifth condition coming fifth, condition end variable for fifth condition on the fifth place under `condition end variables:`).
The parameters are the following and should be printed in this order:

```
number of conditions:
experimentname:
variables:
conditionnames:
colors:
zero:
condition begin variables:
condition begin properties:
condition end variables:
condition end properties:
```

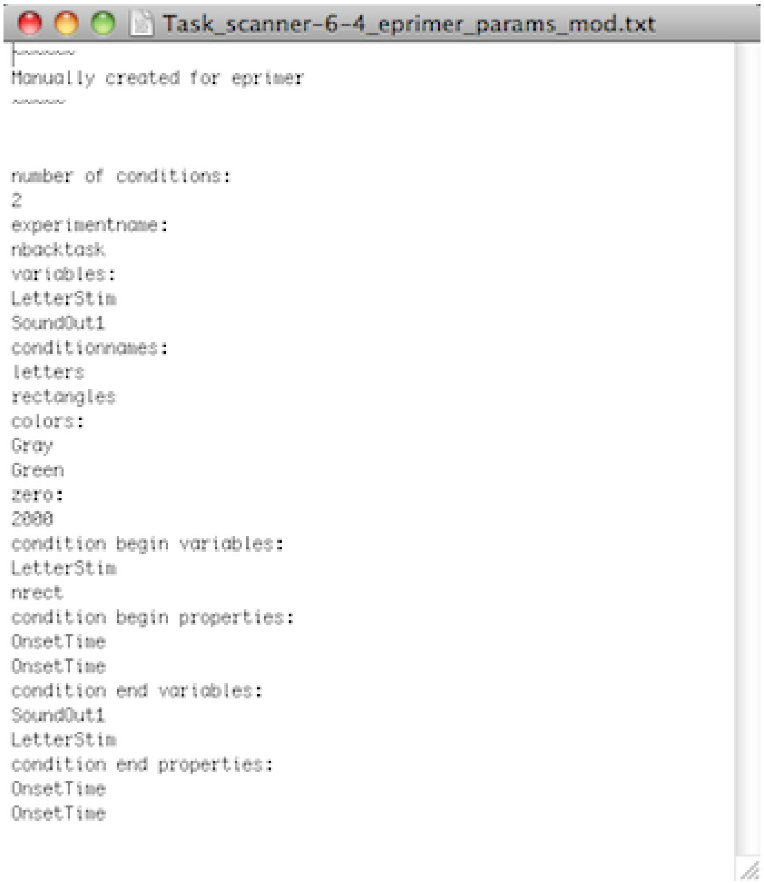
The color names that can be used, exactly as they are printed below, are:

```
Gray
Black
Red
Green
Blue
White
Yellow
Pink
Lightblue
Burgundy
Darkgreen
Darkblue
Orange
Brightblue
Pinkred
```

This text file should be re-usable for all subjects of one experiment. The order of occurrence of the variables in the E-Prime Log file will determine the final form of the stimulation protocol per subject.

Example

Imagine, for example, that we would like to have a condition in the BrainVoyager stimulation protocol with the name “letters”. We write then in the configuration file under `conditionnames:` “letters” (without quotes). Also, we know that the begin of each interval for that condition is defined in the E-Prime Log text by “LetterStim.OnsetTime” and the end of an interval for this condition is defined by “SoundOut1.OnsetTime”. Then, for this condition the variable that `eprimer` should look for is “LetterStim” and we write “LetterStim” (without quotes) under `condition begin variables:` in the configuration file.
The property of LetterStim that contains the start time is “OnSetTime”. Hence we write once “OnsetTime” under `condition begin properties:`. For the end of the interval of the ‘letters’ condition, we take “SoundOut1” for `condition end variables:` and “OnsetTime” for `condition end properties`; although this is again the same property name as for the begin of the interval, this property belongs to “SoundOut1” in this case. Since we would like to have this condition in the color gray, we look in the abovementioned list and write under `Gray` under `colors:`.
An example parameter file is shown in figure 5.

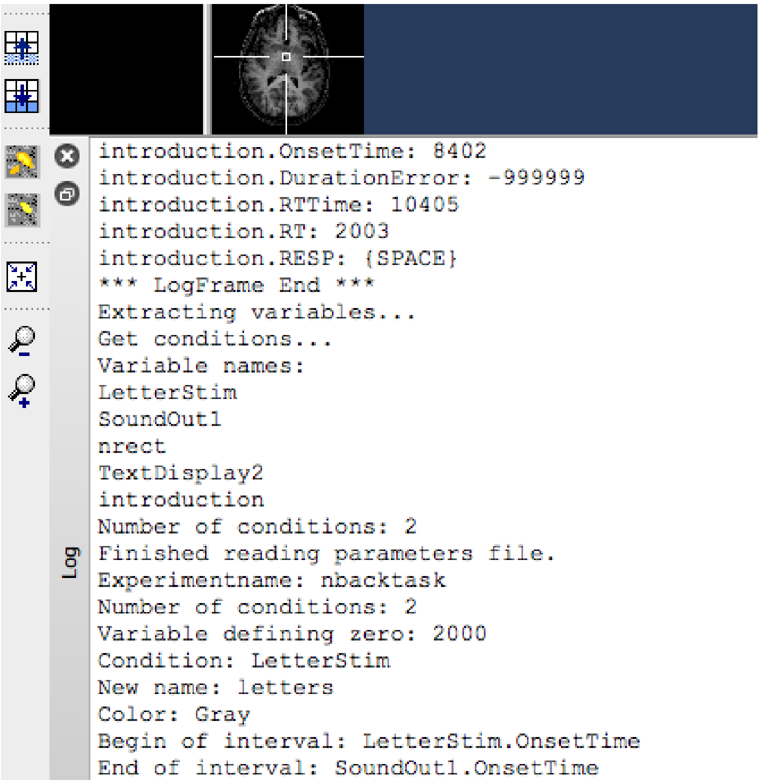


```
~~~~~~
Manually created for eprimer
~~~~~


number of conditions:
2
experimentname:
nbacktask
variables:
LetterStim
SoundOut1
conditionnames:
letters
rectangles
colors:
Gray
Green
zero:
2000
condition begin variables:
LetterStim
nrect
condition begin properties:
OnsetTime
OnsetTime
condition end variables:
SoundOut1
LetterStim
condition end properties:
OnsetTime
OnsetTime
```

Figure 5: E-Prime example parameter file for another experiment with 2 conditions

2.4 Convert experiment log to stimulation protocol

After the button “Convert experiment log to stimulation protocol” on the `eprimer` dialog has been clicked, `eprimer` will extract the start and end for the intervals of the selected conditions from the E-Prime Log file. Variables are printed to the BrainVoyager QX Log tab (see figure 6). No user input is required in this stage.



```
introduction.OnsetTime: 8402
introduction.DurationError: -999999
introduction.RTTime: 10405
introduction.RT: 2003
introduction.RESP: {SPACE}
*** LogFrame End ***
Extracting variables...
Get conditions...
Variable names:
LetterStim
SoundOut1
nrect
TextDisplay2
introduction
Number of conditions: 2
Finished reading parameters file.
Experimentname: nbacktask
Number of conditions: 2
Variable defining zero: 2000
Condition: LetterStim
New name: letters
Color: Gray
Begin of interval: LetterStim.OnsetTime
End of interval: SoundOut1.OnsetTime
```

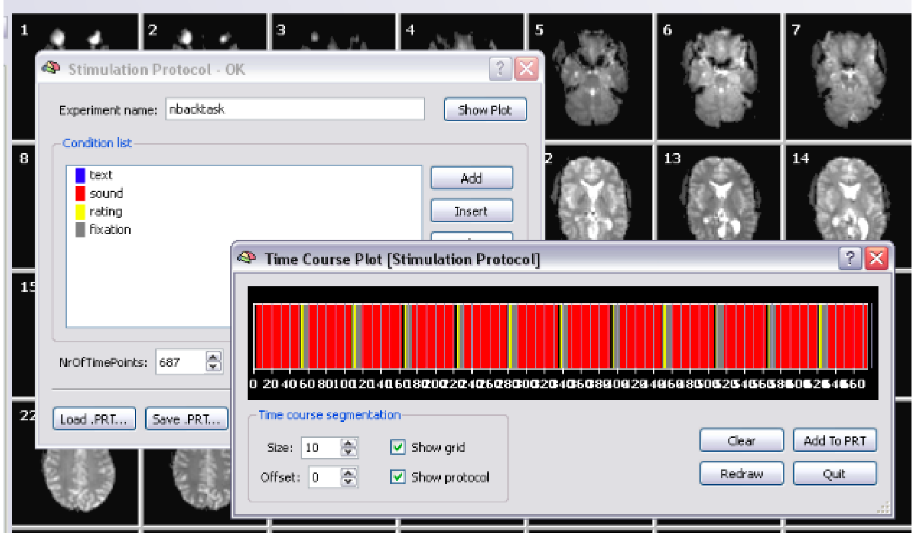
Figure 6: Information printed to the BrainVoyager QX Log tab

Then, the name for the BrainVoyager stimulation protocol file (*.prt) will be formed from the experiment name in the configuration file. The resulting stimulation protocol can be checked via the BrainVoyager “Analysis” ↦ “Stimulation Protocol...” option. On the Stimulation Protocol dialog is a button that will allow to plot the intervals (see figure 7).

The stimulation protocol is a text file which can also be checked via for example Wordpad on Windows, TextEdit on Mac or Kate on Linux (see figure 8).

2.5 Note

In case the BrainVoyager QX Log tab shows empty lines when importing via `eprimer`, please read the following. The cause is a text mark-up problem. The solution for importing the *.txt log file from E-Prime is to copy-paste the logtext to a mark-up free text/programming editor (like Code::Blocks, see http://www.codeblocks.org/), save in a different extension and import the ’mark-up free’ text with `eprimer`.



Experiment name: nbacktask

Condition list

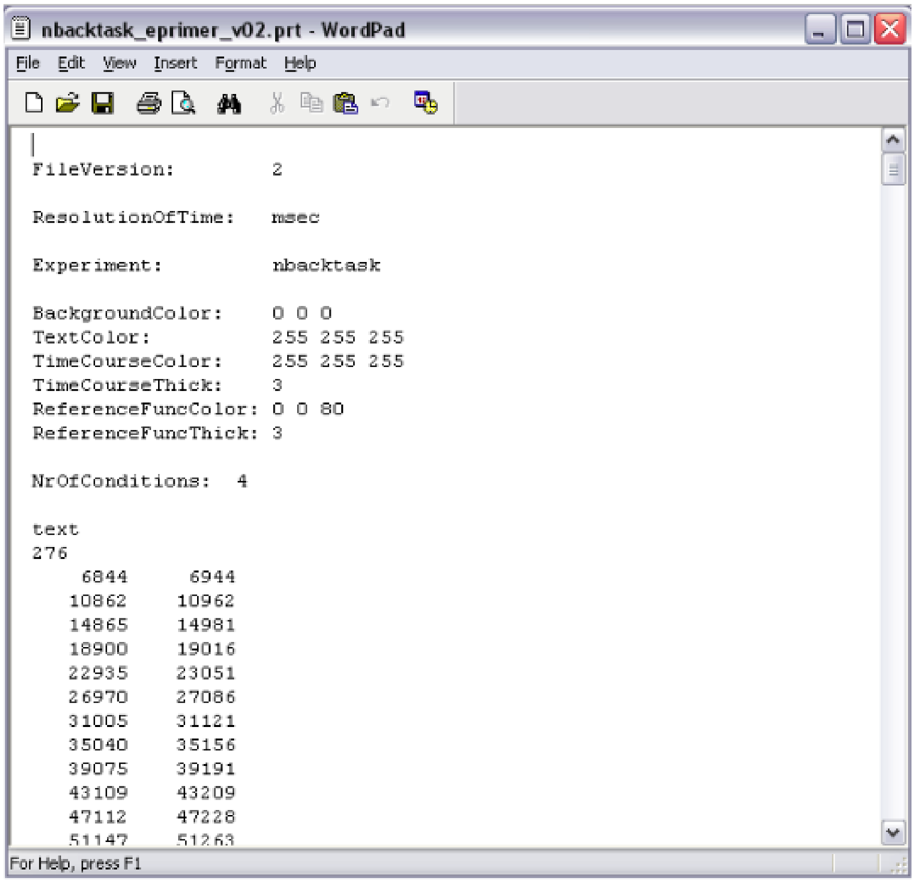
- 🟦 text
- 🟥 sound
- 🟨 rating
- ⬜ fixation

NrOfTimePoints: 687

Size: 10 ☑ Show grid

Offset: 0 ☑ Show protocol

Figure 7: The resulting BrainVoyager stimulation protocol

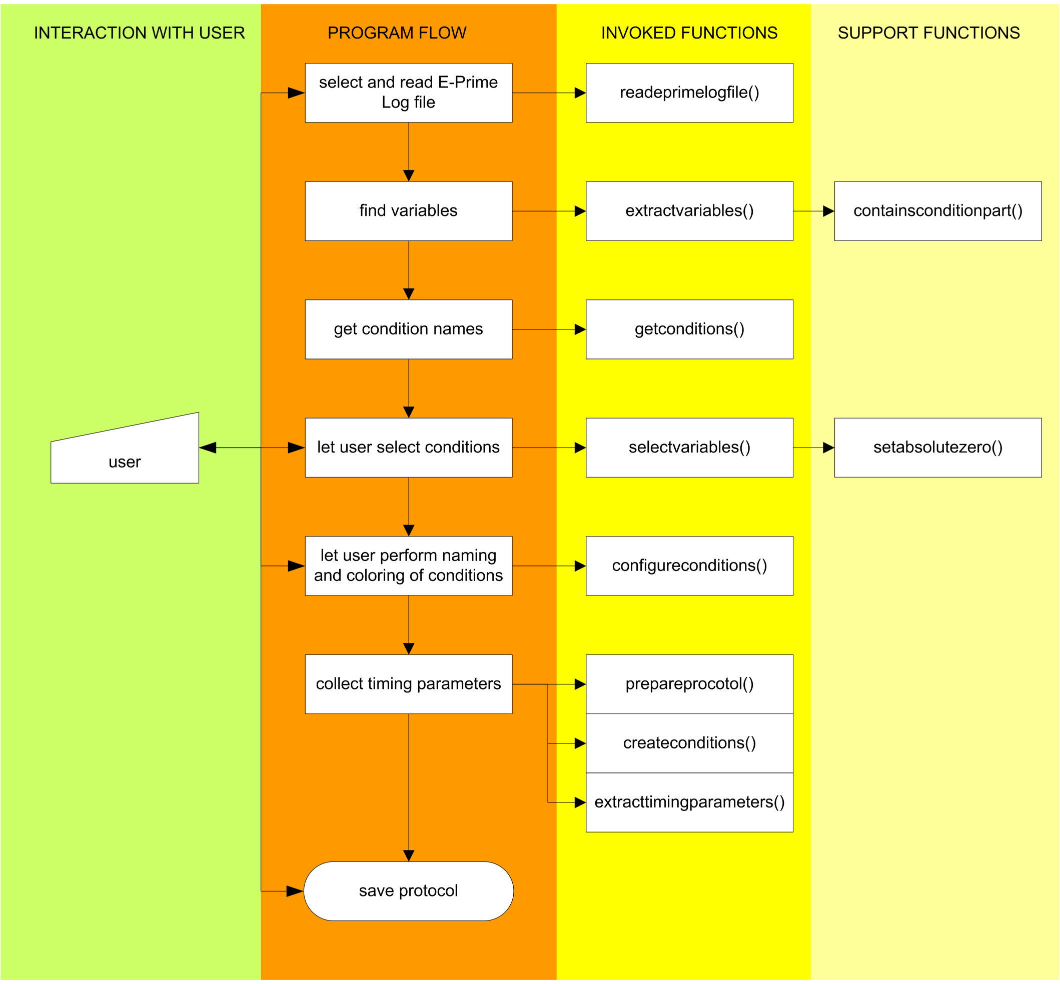

nbacktask_eprimer_v02.prt - WordPad
File Edit View Insert Format Help
For Help, press F1

```
FileVersion:        2

ResolutionOfTime:   msec

Experiment:         nbacktask

BackgroundColor:    0 0 0
TextColor:          255 255 255
TimeCourseColor:    255 255 255
TimeCourseThick:    3
ReferenceFuncColor: 0 0 80
ReferenceFuncThick: 3

NrOfConditions:   4

text
276
    6844     6944
   10862    10962
   14865    14981
   18900    19016
   22935    23051
   26970    27086
   31005    31121
   35040    35156
   39075    39191
   43109    43209
   47112    47228
   51147    51263
```

Figure 8: The resulting BrainVoyager stimulation protocol file

3 Design

The design of `eprimer` is shown in diagram 3 (read from top to bottom). (Please note, the design flowchart is outdated).

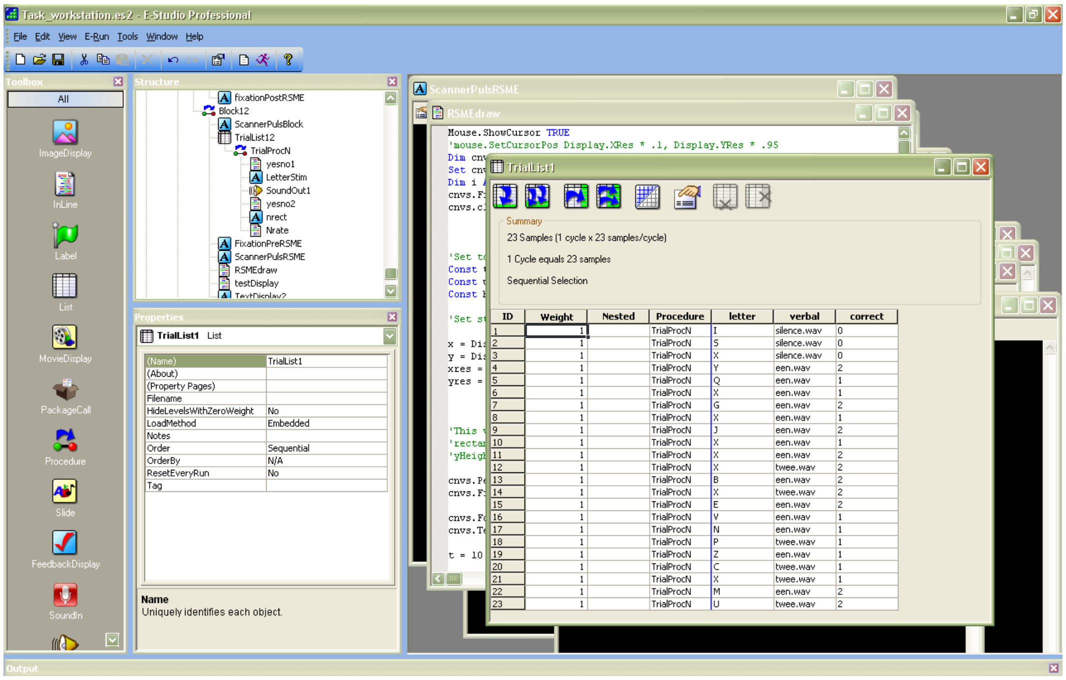


A Logging the timing parameters in E-Prime

In the 'Structure' window of figure 9 one can see which variables are shown at which time point (where the time axis runs from top to bottom). One can select the logging of the variables via right-clicking on the variable names and selecting one or several of 'OnsetDelay', 'OnsetTime', 'DurationError', 'RTTime', 'ACC', 'RT', 'RESP' and 'CRESP'. Please not that if 'Logging' is selected, but no properties in particular, nothing will actually be saved to the Log file!

A.1 Notes to take into account before starting the experiment

- In particular, take care that a variable is created and enabled to log the first scanner pulse (in this example `ScannerPulsStart`), to obtain a clear reference point for the timing parameters like RTTime.
- If your stimulation program contains intervals between objects, that in turn are triggered by a scanner pulse, do not take intervals that are (multiples of) TR times. The reason for this is that sometimes the program takes a few milliseconds more or less, which might uncontrollably shift your protocol depending on if the program just catches the scanner pulse or has to wait for the next one.



(Name)	TrialList1
(About)	
(Property Pages)	
Filename	
HideLevelsWithZeroWeight	No
LoadMethod	Embedded
Notes	
Order	Sequential
OrderBy	N/A
ResetEveryRun	No
Tag	

Name
Uniquely identifies each object.

```
Mouse.ShowCursor TRUE
'mouse.SetCursorPos Display.XRes * .1, Display.YRes * .95
Dim cn
Set cn
Dim i A
cnvs.F
cnvs.c


'Set t
Const
Const
Const

'Set s

x = Di
y = Di
xres =
yres =


'This
'recta
'yHeig

cnvs.P
cnvs.F

cnvs.F
cnvs.T

t = 10
```

Summary

23 Samples (1 cycle x 23 samples/cycle)

1 Cycle equals 23 samples

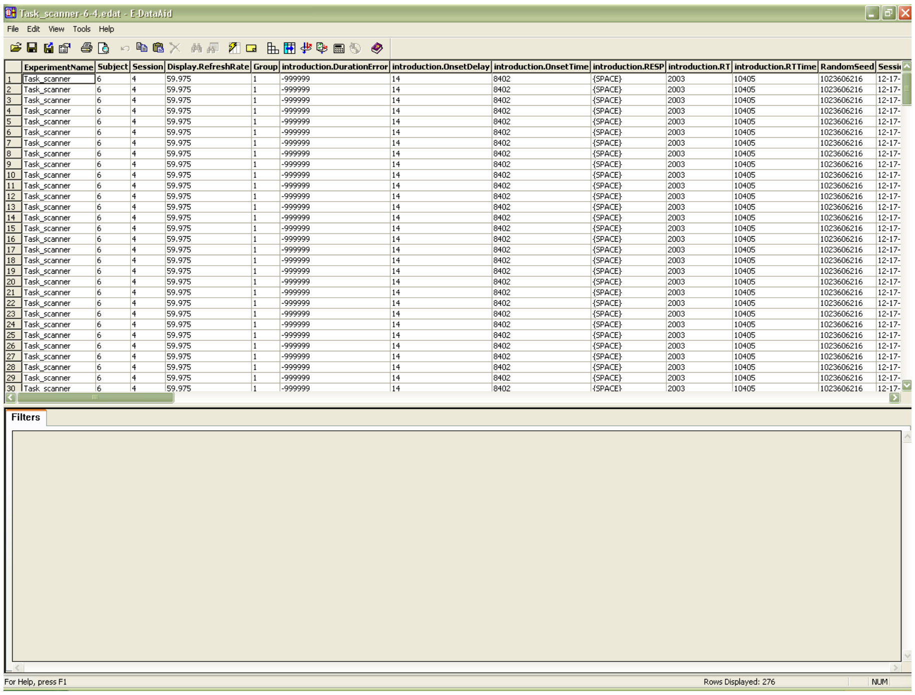
Sequential Selection

ID	Weight	Nested	Procedure	letter	verbal	correct
1	1		TrialProcN	I	silence.wav	0
2	1		TrialProcN	S	silence.wav	0
3	1		TrialProcN	X	silence.wav	0
4	1		TrialProcN	Y	een.wav	2
5	1		TrialProcN	Q	een.wav	1
6	1		TrialProcN	X	een.wav	1
7	1		TrialProcN	G	een.wav	2
8	1		TrialProcN	X	een.wav	1
9	1		TrialProcN	J	een.wav	2
10	1		TrialProcN	X	een.wav	1
11	1		TrialProcN	X	een.wav	2
12	1		TrialProcN	X	twee.wav	2
13	1		TrialProcN	B	een.wav	2
14	1		TrialProcN	X	twee.wav	2
15	1		TrialProcN	E	een.wav	2
16	1		TrialProcN	V	een.wav	1
17	1		TrialProcN	N	een.wav	1
18	1		TrialProcN	P	twee.wav	1
19	1		TrialProcN	Z	een.wav	1
20	1		TrialProcN	C	twee.wav	1
21	1		TrialProcN	X	twee.wav	1
22	1		TrialProcN	M	een.wav	2
23	1		TrialProcN	U	twee.wav	2

Figure 9: One can select the logging of the variables via right-clicking on the variable names

A.1.1 Alternatives

It is also possible to export the logged stimuli timing to Excel format via E-Data Aid (see figure 10). On Windows, this offers the possibility to access Excel (for example using Matlab) via the Component Object Model (COM) and directly script the BrainVoyager stimulation protocol (see the manual “Scripting BrainVoyager from Matlab” on the BrainVoyager wiki).



	ExperimentName	Subject	Session	Display.RefreshRate	Group	introduction.DurationError	introduction.OnsetDelay	introduction.OnsetTime	introduction.RESP	introduction.RT	introduction.RTTime	RandomSeed	Sessi
1	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
2	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
3	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
4	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
5	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
6	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
7	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
8	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
9	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
10	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
11	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
12	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
13	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
14	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
15	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
16	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
17	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
18	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
19	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
20	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
21	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
22	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
23	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
24	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
25	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
26	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
27	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
28	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
29	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-
30	Task_scanner	6	4	59.975	1	-999999	14	8402	{SPACE}	2003	10405	1023606216	12-17-

For Help, press F1   Rows Displayed: 276   NUM

Figure 10: The logged times can also be exported to Excel via E-Data Aid

B Reference time point

E-Prime records the experiment time starting at the beginning of the script execution, when you click "run".
This means that at the time of execution of the object logging the first scanner pulse, the time will be at a value representing this time.

B.1 Example:

After starting the experiment by clicking run, you enter the session numbers et cetera which takes around 5-10 seconds. It takes you another 5-10 seconds to start the scanner. Hence, the timing value recorded in the Eprime log file for the time of the first scanner pulse is anywhere between 10 and 20 seconds! If you use instruction screens or similar things, this time will be even higher.

`eprimer` sets the timing to zero at the given reference point, which can be indicated via the "`zero:`" parameter in the configuration file. To find the right value for this field, you have to look up the correct time in your Eprime log file. It is usually the RTT of the object logging the first scanner pulse (See also Synchronizing your experiment with the scanner, section C).
This value might change every time you run the experiment, so you want to look it up in your Eprime log file to be sure. You will have to update it in the text file which stores your parameter configuration as well, for every run. Using a text editor such as PSPad, search for "zero:". The value following this is the one that you have filled in the "first scanner pulse" field when you first created this parameter file.
Additionally, you need to take your skipped volumes into account as well. As BrainVoyager starts counting from the first unskipped volume on, you need to adjust your experiment timing accordingly. Thus, if you e.g. skip 3 volumes of 2000 ms each = 6000ms; you need to add this time to your reference point.

B.2 Example:

25000ms in the execution of your Eprime script, the scanner starts to collect the first volume (as logged by the first scanner pulse object). Your TR is 2000ms and you discard the first three volumes. Thus, BrainVoyager starts counting with zero at 25000ms + 6000ms = 31000ms Eprime script time. Therefore, 31000ms is the time you have to enter in the "first scanner pulse" field.

C Synchronizing your Experiment with the Scanner

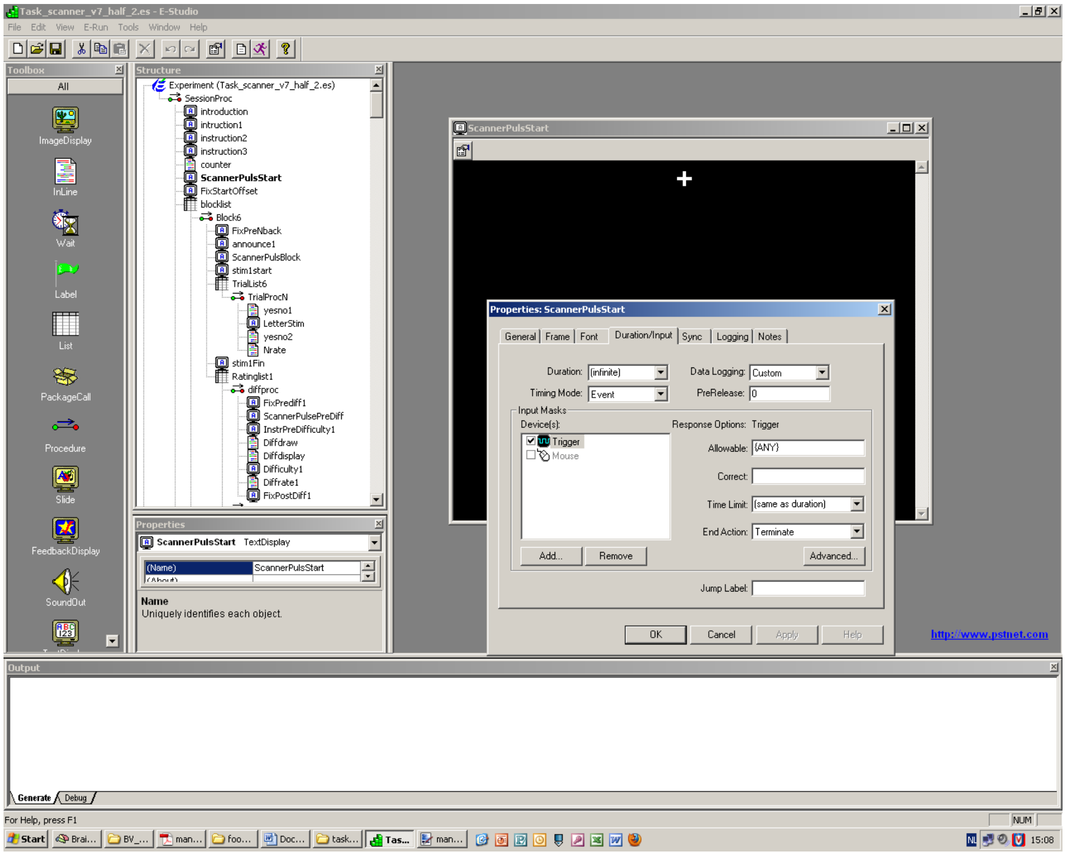
C.1 Creating a Scanner.pulse.start object

Ideally you have an indicator object in your Eprime script that logs the first scanner pulse (see figure 11). This can be done in the following way:

Make sure your scanner has a trigger port, which delivers a pulse via a serial port at every TR start. Ask your MTA or manufacturer for details.

Create a trigger port device in Eprime. In the main screen of E-Studio go to `Edit > Experiment > Devices`. Add a "Port object", click "Edit" and configure it with the values for your system (values shown here are just for demonstration).

Then, create a Text.Object and add the trigger port object as input device and configure so that it is terminated as a reaction on the scanner pulse. Dont forget to turn on the appropriate logging values. This method can be used to ensure correct synchronization.



Properties: ScannerPulsStart

General | Frame | Font | Duration/Input | Sync | Logging | Notes

Duration: (infinite)
Timing Mode: Event
Data Logging: Custom
PreRelease: 0

Input Masks
Device(s):
☑ Trigger
☐ Mouse

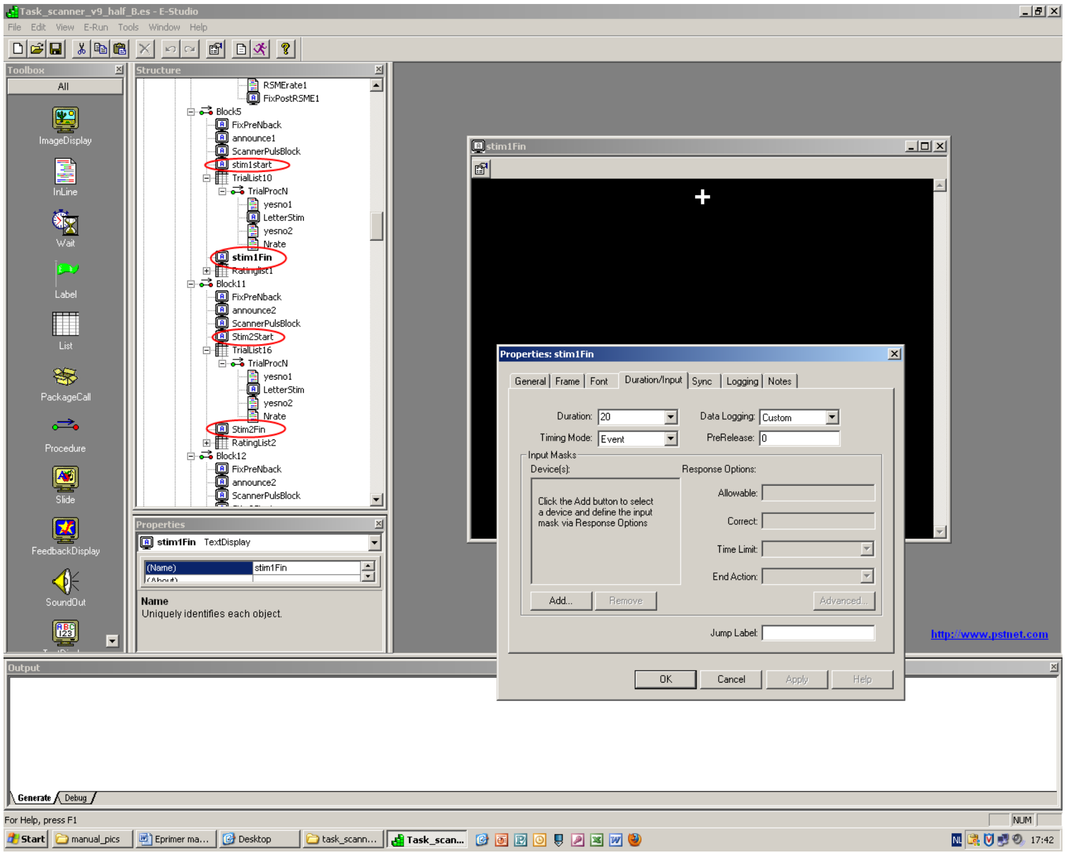
Response Options: Trigger
Allowable: {ANY}
Correct:
Time Limit: (same as duration)
End Action: Terminate
Jump Label:

Figure 11: Scanner pulse object

D Eprimer FAQ

D.1 I have to sort out the order of my conditions by hand, as I use identical objects for different conditions in my Eprime script. Is there anything I can do to make this easier?

You want to keep your program lean by not including differently named copies of identical objects for every condition. Instead, you can use identical objects that you “feed” with different contents from list files. In Eprimer04 however, those variables have identical names, thus making it impossible to separate them. This can be remedied by using small indicator objects (see figure 12), lasting for example 20ms. Just give them logical names. We have a stimulation procedure that we use in 3 different conditions, employing the same objects. By placing the small indicators before and after the actual procedure, we can use their timing values instead.



Task_scanner_v9_half_B.es - E-Studio

Structure:

```
RSMErate1
FixPostRSME1
Block5
  FixPreNback
  announce1
  ScannerPulsBlock
  stim1start
  TrialList10
    TrialProcN
      yesno1
      LetterStim
      yesno2
      Nrate
  stim1Fin
  Ratinglist1
Block11
  FixPreNback
  announce2
  ScannerPulsBlock
  Stim2Start
  TrialList16
    TrialProcN
      yesno1
      LetterStim
      yesno2
      Nrate
  Stim2Fin
  RatingList2
Block12
  FixPreNback
  announce2
  ScannerPulsBlock
```

Properties: stim1Fin TextDisplay

(Name): stim1Fin

Name
Uniquely identifies each object.

Properties: stim1Fin — Duration/Input

Duration: 20
Timing Mode: Event
Data Logging: Custom
PreRelease: 0

Input Masks
Device(s): Click the Add button to select a device and define the input mask via Response Options

Figure 12: Indicator objects

D.2 The BVQXLog tab shows empty lines when importing via eprimer

The cause is a text mark-up problem. The solution for importing the *.txt log file from E-Prime is to copy-paste the logtext to a mark-up free text editor (like OpenOffice Writer), save in a different extension and import the ’mark-up free’ text with `eprimer`. Instead of copy-paste, import as generic Unicode can help as well.

D.3 I do not see all of my conditions listed

Make sure that the objects defining your conditions do not have too similar names. In order to prevent double variables, Eprimer04 omits variables it deems identical. We found that the current version can handle numbered variables (xy1, xy2) but chokes on instances where a certain name is used with different prefixes. For example, one condition in our experiment was called difficulty1, and we called the fixation screen text object prior to it FixPreDifficulty1, which made Eprimer omit difficulty1. If you have this problem, simply rename the variable name in a copy of your .txt log files with a text editor such as PSPad.

D.4 My stimulation protocol is NOT OK

In most cases, this is caused by overlapping conditions, either due to missed responses or due to errors in defining the conditions. Eprime logs RTs and RTTs for missed responses as 0. Thus, if you use a reaction time as a defining parameter, in case of a missed response you will have a trial that lasts from the negative value of your starting reference point (e.g. -31000ms) until the indicated end of that trial, naturally overlapping everything in between. Hint: If you suspect this, look for the negative time of that protocols reference time point with the search function of any text editor.

If this is not the problem, there is probably an error in the definition of the conditions.

Narrow done all possible errors by copying parts of your protocol file into an empty tryout .prt text file and inspecting them in BrainVoyager. Add the conditions/fixations piece by piece. As soon as BrainVoyager switches to NOT OK, you have one part of the problem. Leave this part in, delete the rest of the conditions from that tryout file and start adding the other conditions again. By this, you can identify which conditions overlap.

You then can proceed to see if there is an error in the parameters entered, or for example a missed response by a subject.