



# Program *BehaPresenter*

Wersja: 1.0.4 (2017.04.20)

Dokument: 2017.04.20

Autor: Paweł M. Boguszewski, [pmbogusz@gmail.com](mailto:pmbogusz@gmail.com)

<http://pmbogusz.net/?a=behapresenter>

## Spis treści

1	informacje ogólne.....	3
2	Uruchamianie programu .....	3
3	Sekwencja komend - plik BPX.....	4
3.1	Komentarze .....	4
3.2	Łączenie plików .....	5
3.2.1	include .....	5
3.3	Komendy ogólne.....	5
3.3.1	verbose .....	5
3.3.2	mouse_hide .....	5
3.3.3	mouse_show .....	5
3.3.4	numpad_on .....	5
3.3.5	numpad_off.....	5
3.3.6	background.....	6
3.3.7	blank .....	6
3.3.8	halt.....	6
3.4	Sterowanie programem.....	6
3.4.1	wait_time.....	6
3.4.2	wait_key .....	6
3.4.3	wait_mouse .....	7
3.4.4	wait_input .....	7
3.5	Stopery i logowanie do logu krótkiego.....	7
3.5.1	timer_set .....	8
3.5.2	timer_log .....	8
3.5.3	log_text.....	8
3.5.4	log_input.....	8
3.5.5	timer_input.....	9
3.5.6	timer_inptxt.....	9
3.6	Punkt fiksacji wzroku .....	10
3.6.1	fix_show.....	10
3.6.2	fix_hide .....	10
3.6.3	fix_color.....	10
3.7	Teksty .....	10
3.7.1	text_font.....	10
3.7.2	text_just.....	11
3.7.3	text_color .....	11
3.7.4	text_bgcol .....	11
3.7.5	text_text .....	11
3.7.6	text_name .....	11
3.7.7	text_size.....	11
3.7.8	text_show .....	12
3.7.9	text_hide.....	12
3.7.10	text_vpos .....	12
3.7.11	text_hpos.....	12

3.7.12	text_vpospx .....	12
3.7.13	text_hpospx.....	12
3.8	Obrazy.....	12
3.8.1	pic_load .....	13
3.8.2	pic_name .....	13
3.8.3	pic_show.....	13
3.8.4	pic_hide .....	13
3.8.5	pic_vpos.....	13
3.8.6	pic_hpos .....	13
3.8.7	pic_vpospx.....	13
3.8.8	pic_hpospx.....	13
3.9	Odtwarzanie i nagrywanie dźwięków.....	14
3.9.1	sound_play .....	14
3.9.2	sound_stop .....	14
3.9.3	beep.....	14
3.9.4	record_start.....	14
3.9.5	record_stop .....	14
3.10	Odtwarzanie filmów - plików wideo.....	15
3.10.1	video_load .....	15
3.10.2	video_play .....	15
3.10.3	video_stop.....	15
3.10.4	video_vpos.....	15
3.10.5	video_hpos .....	15
3.10.6	video_vpospx.....	15
3.10.7	video_hpospx .....	15
3.10.8	video_width .....	15
3.10.9	video_height.....	16
3.11	Pozycjonowanie obiektów na ekranie.....	17
3.11.1	Pozycjonowanie względne .....	17
3.11.2	Pozycjonowanie bezwzględne.....	18
3.12	Kolory.....	19
3.12.1	Definiowanie kolorów .....	19
3.12.2	Dopasowanie koloru tła.....	19
4	Pliki wynikowe - logi .....	20
4.1	Plik BPS .....	20
5	Dodatki .....	22
5.1	Przydatne program.....	22
5.1.1	FastStone Image Viewer .....	22
5.1.2	Notepad++.....	22
5.1.3	Color Seizer.....	22
5.1.4	Audacity.....	22
5.2	Kody klawiszy.....	23
5.3	Tabela kolorów .....	25

## 1 informacje ogólne

Program BehaPresenter służy do prowadzenia doświadczeń psychofizjologicznych i neuropsychologicznych. Umożliwia prezentację bodźców przy pomocy komputera z jednoczesną rejestracją odpowiedzi osoby badanej.

Program wczytuje listę poleceń z pliku tekstowego i wykonuje te komendy sekwencyjnie. Polecenia dotyczą przygotowania i prezentacji bodźców oraz oczekiwania na reakcję osoby badanej. Jest to lista rozkazów, które ma wykonać komputer w trakcie badania.

Program może prezentować następujące bodźce:

- Tekst – dowolna wielkość, kolor czcionki i tła, i położenie na ekranie.
- Obrazy z plików (bmp, jpg, png)
- Krótki sygnał dźwiękowy (tzw. beep)
- Pliki dźwiękowe zapisane w formacie WAV
- stały punkt fiksacji wzroku.

Dane zbierane:

- Wszelka aktywność klawiatury - zarówno czas wciśnięcia jak i zwolnienia klawiszy, rejestracja wielu klawiszy równocześnie.
- Kliknięcia myszą (dokładna pozycja, nazwa klikniętego bodźca, wciśnięty klawisz)
- Nagrany dźwięk z mikrofonu komputera – odpowiedzi ustne osoby badanej.

Wynikiem działania programu jest plik tekstowy zawierający spis wszystkich „zdarzeń” – definiowania i wyświetlania bodźców oraz kliknięć w klawiaturę i wciśnięć myszy. Czas zdarzeń jest zapisywany z dokładnością do milisekund.

## 2 Uruchamianie programu

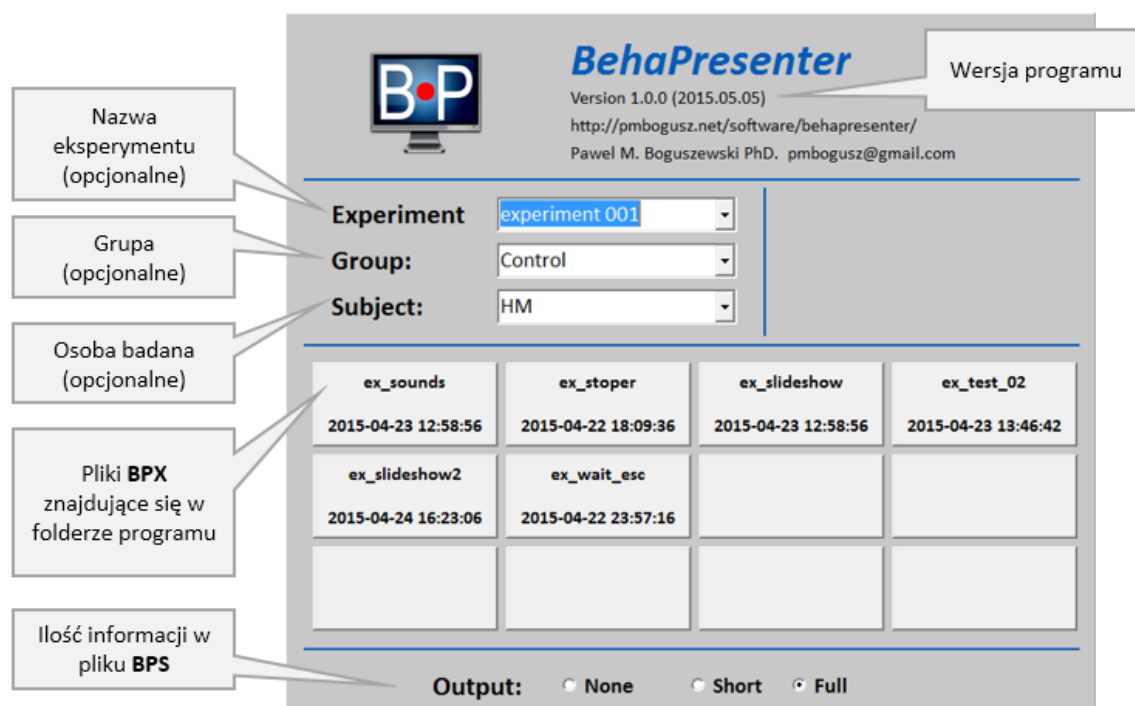
Program nie wymaga instalacji, można uruchamiać go bezpośrednio z nośników typu pendrive. Aby uruchomić program wystarczy dwukrotnie kliknąć plik "BehaPresenter.exe".

Po uruchomieniu program wyświetla ekran startowy. Po opcjonalnym uzupełnieniu pól `Experiment`, `Group` i `Subject` wystarczy wcisnąć przycisk odpowiadający plikowi **BPX**, co uruchomi eksperyment.

Pliki z komendami (eksperymenty) z rozszerzeniem **BPX** powinny znajdować się w tym samym katalogu co program.

Zamiast ręcznego uzupełniania pól `Experiment`, `Group` i `Subject` można je wcześniej zdefiniować w plikach tekstowych o nazwach, odpowiednio:

`experiments.bpl`, `groups.bpl`, `subjects.bpl` - wtedy wpisane w pliki nazwy będą dostępne jako rozwijane menu.



### 3 Sekwencja komend - plik BPX

Przebieg eksperymentu, sekwencje komend, zapisujemy w pliku tekstowym przy pomocy Notatnika (Notepad) lub programu Notepad++ (patrz 5.1.2). Plik powinien mieć rozszerzenie **.bpx**. Po utworzeniu pliku tekstowego możemy zapisać go jako BPX lub zmienić jego nazwę przy użyciu klawisza F2.

W tym pliku definiujemy bodźce, które chcemy zaprezentować osobie badanej i ustalamy ich sekwencje wraz z komendami sterującymi - oczekiwaniem na reakcje osoby badanej.

Dla przyspieszenia działania programu i klarowności zapisu najwygodniej definiować bodźce na początku pliku komend. Pozwala to uniknąć ewentualnych opóźnień związanych z wczytywaniem plików graficznych do pamięci programu.

#### 3.1 Komentarze

Każdy tekst w linii po znaku średnika „;” lub dwóch ukośników „//” jest ignorowany. Pozwala to umieszczać w pliku komend komentarze, opisywać poszczególne komendy lub wyłączać je bez konieczności kasowania całych linii. W przykładzie poniżej tekst zielony będzie ignorowany.

```
; to jest komentarz
// to też komentarz
verbose 2      // włącz pełne logowanie
mouse_hide    ; ukryj kursor myszy
; blank       // ta komenda nie zostanie wykonana!
```

**UWAGA:** Nie należy stosować znaków ; w opisie stoperów, tekstach do wyświetlenia i w komendzie **log\_text** - wszystko co znajdzie się po : będzie ignorowane

## 3.2 Łączenie plików

### 3.2.1 include

**include** <file>

wstawia w miejscu wystąpienia komendy **include** inny plik z komendami.

Jeżeli chcemy uruchomić różne warianty doświadczenia, ale z tym samym zestawem bodźców to możemy bodźce zdefiniować w osobnym pliku i „wczytywać” je do plików BPX. Pozwala to ograniczyć długość kodu.

## 3.3 Komendy ogólne

### 3.3.1 verbose

Określenie informacji, jakie znajdą się w pliku wynikowym **BPS**. Komenda **verbose** w pliku BPS ma wyższy priorytet niż poprzednie ustawienie - zmiana zmiennej Output na ekranie początkowym lub parametr w skrócie do programu.

**verbose 0** Program nie zapisuje ani pliku **BPS** ani dźwięków. Stosowane do testów, kiedy nie chcemy kasować plików tymczasowych.

**verbose 1** Program zapisuje do pliku **BPS** nagłówki i wyłącznie krótki LOG

**verbose 2** Program zapisuje do pliku **BPS** nagłówki, krótki LOG i pełen log ze wszystkimi informacjami. Ponieważ oba logi są tworzone równolegle, to informacja z logu krótkiego pojawi się również w logu długim. To jest ustawienie domyślne.

### 3.3.2 mouse\_hide

**mouse\_hide**

ukrywa kursor myszy na ekranie - mysz należy usuwać kiedy oczekujemy wyłącznie przyciskania klawiszy

### 3.3.3 mouse\_show

**mouse\_show**

pokazuje kursor myszy na ekranie - mysz trzeba włączyć przed komendami, kiedy oczekujemy na kliknięcie obiektu na ekranie.

### 3.3.4 numpad\_on

**numpad\_on**

Włącza NumLock – klawiatura numeryczna będzie działać jak kalkulator – wysyłać cyfry w postaci kodów klawiszy NUMPAD0, NUMPAD1 - patrz 5.2

### 3.3.5 numpad\_off

**numpad\_off**

Wyłącza NumLock – klawiatura numeryczna będzie działać jako klawisze nawigacji, będzie wysyłać UP, DOWN, HOME - patrz 5.2

**UWAGA:** używając komend włączających i wyłączających klawiaturę numeryczną trzeba pamiętać o wyłączeniu komunikatów systemowych (pokazywania na ekranie, że został wciśnięty NumLock) – jest to specyficzne sprzętowo – różne dla każdego komputera lub sterownika klawiatury. Bardzo często występuje w laptopach i w klawiaturach Logitech. Można też komendę tę użyć dla pewności na samym początku, przed ekranem z informacją o badaniu

### 3.3.6 background

**background <kolor>**

Ustawia kolor tła ekranu. **<kolor>** jest kodem lub nazwą koloru – patrz 3.11. kolor ekranu można zmienić w dowolnym momencie prezentacji.

### 3.3.7 blank

**blank**

czyści ekran - ukrywa wszystkie obrazki, teksty i znak fiksacji. Jeżeli chcemy ukryć pojedynczy element to należy zastosować odpowiednią komendę **\_hide**

### 3.3.8 halt

**halt**

przerywa i kończy działanie programu. Komenda nie jest wymagana, program sam zakończy swoje działanie po dojściu do końca listy komend.

## 3.4 Sterowanie programem

### 3.4.1 wait\_time

**wait\_time <czas>**

Przed wykonaniem kolejnej komendy program czeka podaną ilość milisekund. Służy do opóźnienia wyświetlenia bodźca. W tym czasie są rejestrowane wszelkie wciśnięcia klawiszy i kliknięcia myszy, ale są ignorowane.

### 3.4.2 wait\_key

**wait\_key [<klawisze>]**

Czeka na wciśnięcie klawisza. Jeżeli parametr **[<klawisze>]** będzie pusty to oczekiwane jest wciśnięcie dowolnego klawisza. Jeżeli zdefiniujemy nazwy oczekiwanych klawiszy, to tylko ich wciśnięcie będzie zapewniało postęp badania.

Lista kodów klawiszy **[<klawisze>]** powinna być rozdzielona **przecinkami**. Spis kodów – patrz 5.2

Wszelkie wciśnięcia klawiszy z poza listy i kliknięcia myszy są rejestrowane (będą zapisane w logu i pliku BPS), ale są ignorowane - program nie wykonuje dalszych poleceń.

```
; Przykład użycia komend typu wait_
```

```
wait_time 3000          // czeka 3 sekundy
wait_key                // czeka na wciśnięcie dowolnego klawisza
wait_key SPACE          // czeka na wciśnięcie spacji
wait_key KEY_A, KEY_F   // czeka na wciśnięcie klawiszy A lub F
wait_key KEY_2          // czeka na wciśnięcie klawisza 2 na głównej
                        // klawiaturze (klawisz nad literami)
wait_key NUMPAD2        // czeka na wciśnięcie klawisza 2 na
                        // klawiaturze numerycznej - numlock musi
                        // być włączony
```

### 3.4.3 wait\_mouse

**wait\_mouse [<obiekty>]**

Czeka na kliknięcie dowolnego przycisku myszy lub, w przypadku podania listy [**<obiekty>**], na kliknięcie zdefiniowanego wcześniej bodźca – obrazku lub tekstu. Wszelkie wciśnięcia klawiszy i kliknięcia myszy poza obiektami z listy są rejestrowane, ale są ignorowane.

Obiekty definiujemy poprzez ich numery przy pomocy identyfikatorów:

picture_001	– obraz o numerze 1
text_001	– tekst o numerze 1
background	– tło

```
; Przykład oczekiwania na kliknięcie obrazka
```

```
pic_load 1 puppy.jpg      // wgraj obraz nr 1 z pliku o nazwie
                          // puppy.jpg
pic_show 1                // wyświetl obraz nr 1
wait_mouse picture_001    // czeka aż osoba badana kliknie w
                          // obrazek nr 1
```

### 3.4.4 wait\_input

**wait\_input**

Czeka na wciśnięcie dowolnego klawisza lub kliknięcie dowolnego przycisku myszy.

## 3.5 Stopery i logowanie do logu krótkiego

Stopery pozwalają na mierzenie odcinków czasu. Są wygodne kiedy nie chcemy analizować długiego (szczegółowego) logu, a jedynie interesują nas pewne odcinki czasu.

Można używać do 999 stoperów – są one identyfikowane są za pomocą numeru. Po rozpoczęciu programu są automatycznie zerowane – ustawione na czas uruchomienia skryptu.

### 3.5.1 timer\_set

**timer\_set <x>** zapamiętuje bieżący czas w stoperze („zeruje” stoper) o podanym numerze. **<x>** jest numerem stopera.

### 3.5.2 timer\_log

**timer\_log <x> <tekst>**

Zapisuje wartość stopera o numerze **<x>** do logu krótkiego pod nazwą **<tekst>**. Czas jest podany w milisekundach które upłynęły od komendy **timer\_set** dla danego stopera. Zapisanie do logu nie resetuje stopera – czas jest liczony dalej.

Przykładowe wykorzystanie komend **timer\_set** i **timer\_log**

```
; Przykład komendy timer_log

timer_set 1          // zeruje stoper nr 1
wait_time 100        // czeka 100 ms
timer_log 1 pierwszy // zapisuje wartość stopera 1
timer_set 2          // zeruje stoper nr 2
wait_time 100        // czeka 100 ms
timer_log 1 pierwszy // zapisuje wartość stopera 1
timer_log 2 drugi    // zapisuje wartość stopera 1
```

Wynikowy plik **BPS**

```
=== [LOG] =====
pierwszy  100
pierwszy  200
drugi     100
=== [LONG LOG] =====
      5      5      timer_set 1
      5      0      wait_time 100
    105    100      timer_log  1      pierwszy    100
    105      0      timer_set 2
    105      0      wait_time 100
    205    100      timer_log  1      pierwszy    200
    205      0      timer_log  2      drugi 100
```

### 3.5.3 log\_text

**log\_text <tekst>**

Zapisuje dowolny tekst w logu krótkim. Może służyć do opisu danych eksperymentalnych, zaznaczania kluczowych miejsc w przebiegu eksperymentu, etc. W logu krótkim nie jest zapisywany czas wykonania tej komendy!

### 3.5.4 log\_input

**log\_input**

Zapisuje kod ostatnio wciśniętego klawisza (patrz kody klawiszy 5.2) lub ostatnio klikniętego obiektu - obrazka lub tekstu. Obiekty są opisywane poprzez ich nazwy (patrz komendy **txt\_name** lub **pic\_name** ) lub numery przy pomocy identyfikatorów:



picture\_001    – obraz o numerze 1  
 text\_001       – tekst o numerze 1  
 background    – tło

### 3.5.5 timer\_input

**timer\_input <x>**

Połączenie **timer\_log** i **log\_input**. Zapisuje kod ostatnio wciśniętego klawisza (patrz kody klawiszy 5.2) lub ostatnio klikniętego obiektu - obrazka lub tekstu wraz z odczytem stopera o numerze **<x>**. Obiekty są opisywane poprzez ich nazwy (patrz komendy **txt\_name** lub **pic\_name**) lub numery przy pomocy identyfikatorów:

picture\_001    – obraz o numerze 1  
 text\_001       – tekst o numerze 1  
 background    – tło

### 3.5.6 timer\_inptxt

**timer\_inptxt <x> <tekst>**

Połączenie **timer\_log**, **log\_input** i **log\_text**. Zapisuje kod ostatnio wciśniętego klawisza (patrz kody klawiszy 5.2) lub ostatnio klikniętego obiektu - obrazka lub tekstu wraz z odczytem stopera o numerze **<x>**. Obiekty są opisywane poprzez ich nazwy (patrz komendy **txt\_name** lub **pic\_name**) lub numery przy pomocy identyfikatorów:

picture\_001    – obraz o numerze 1  
 text\_001       – tekst o numerze 1  
 background    – tło

Dodatkowo zapisuje **<tekst>**, można stosować jako znacznik oczekiwanych odpowiedzi w logu krótkim.

Przykładowe wykorzystanie komend **log\_input** i **timer\_input**

```

; Przykład komend log_input i timer_input

timer_set 1           // zeruje stoper nr 1
text_show 1           // pokaż bodziec na ekranie
wait_input            // oczekiwanie na wciśnięcie klawisza
                      // lub klik myszy
log_input              // zapisuje do logu krótkiego kod klawisza
                      // lub kliknięty obiekt
timer_log 1 pierwszy  // zapisuje wartość stopera 1 z tekstem
                      // "pierwszy" do logu krótkiego
log_text Odpowiedz    // Zapisuje tekst "odpowiedz" do logu krótkiego
timer_input 1          // zapisuje do logu krótkiego wartość stopera o
                      // numerze 1 i kod ostatnio wciśniętego
                      // klawisza lub kliknięty obiekt
  
```

Wynikowy plik **BPS** - osoba kliknęła w tekst nr 1 lewym klawiszem myszy.

```
=== [LOG]=====
text_001
pierwszy 1253
Odpowiedz
text_001 1253
=== [LONG LOG]=====
10      10      text_font 1 Arial
17      7      text_color 1 green
17      0      text_bgcolor 1 silver
17      0      text_size 1 200
23      6      text_text 1 TEST
23      0      timer_set 1
23      0      text_show 1 546 391 830 299
40      17     wait_input
1276    1237   mouse text_001 Left L 429 234
1276    0      text_001
1276    0      timer_log 1 pierwszy 1253
1276    0      Odpowiedz
1276    0      timer input 1 text 001 1253
```

## 3.6 Punkt fiksacji wzroku

### 3.6.1 fix\_show

**fix\_show**

Pokazuje predefiniowany punkt fiksacji wzroku - krzyż na środku ekranu. Jeżeli chcemy inny rodzaj punktu fiksacji lub chcemy zdefiniować jego lokalizację najlepiej użyć bodźca tekstowego (patrz 3.7).

### 3.6.2 fix\_hide

**fix\_hide**

Ukrywa predefiniowany punkt fiksacji wzroku

### 3.6.3 fix\_color

**fix\_color <kolor>**

Ustawia kolor punktu fiksacji wzroku. **<kolor>** jest kodem lub nazwą koloru – patrz 5.3

## 3.7 Teksty

Bodźce tekstowe są podstawową formą prezentowania treści osobie badanej. Można stosować do 999 predefiniowanych bodźców, dowolnie manipulować wielkością czcionki, krojem, kolorem czcionki i tła oraz pozycją.

W poniższych komendach **<x>** oznacza numer bodźca tekstowego – od 1 do 999

### 3.7.1 text\_font

**text\_font <x> <nazwa>**

Ustawia krój czcionki – nazwa powinna być nazwą taką, jaką czcionka ma w systemie Windows, np. **Calibri**, **Arial** etc.

Domyślną czcionką jest Calibri – zawsze trzeba sprawdzić działanie tej komendy, ponieważ jej skuteczność zależy od obecności i konfiguracji fontów w systemie.

### 3.7.2 text\_just

**text\_just <x> <side>**

Ustawia rodzaj wyrównania tekstu w przypadku tekstów wielolinijkowych (zawierających znacznik <br>). **<side>** określa typ wyrównania tekstu:

left	– do lewej strony
right	– do prawej strony
center	– do środka

Nie ma możliwości wyrównywania tekstów do obu stron (justyfikacja). Szerokość okna tekstu jest tworzona automatycznie na podstawie wielkości czcionki i długości tekstu.

### 3.7.3 text\_color

**text\_color <x> <kolor>**

Ustawia kolor czcionki. **<kolor>** jest kodem lub nazwą koloru – patrz 5.3

### 3.7.4 text\_bgcol

**text\_bgcol <x> <kolor>**

Ustawia kolor tła czcionki. Domyślnie tekst nie posiada tła (tło jest przezroczyste) ale można je zdefiniować. **<kolor>** jest kodem lub nazwą koloru – patrz 5.3.

Jeżeli chcemy usunąć tło należy wpisać jako parametr **<kolor>** „none”

### 3.7.5 text\_text

**text\_text <x> <tekst>**

Ustawia tekst, który ma być pokazany. **<tekst>** jest dowolnym łańcuchem znaków w jednej linii.

Tekst może być długi, zostanie automatycznie dopasowany do wielkości okienka tekstowego a jego nadmiar przycięty. Jeżeli chcemy wprowadzić w tekście nową linię to należy wpisać znacznik **<br>**

### 3.7.6 text\_name

**text\_name <x> <tekst>**

Nadaje nazwę obiektowi. **<tekst>** jest dowolną nazwą - łańcuchem znaków w jednej linii. Jeżeli nadamy obiektowi nazwę to będzie ona używana w logu krótkim w komendzie **log\_input** i **timer\_input** zamiast identyfikatora.

Nazwy nie muszą być unikalne (mogą się powtarzać) – służą jedynie do wygodniejszego logowania zdarzeń w logu krótkim

### 3.7.7 text\_size

**text\_size <x> <rozmiar>**

Rozmiar czcionki w punktach.

### 3.7.8 text\_show

**text\_show <x>** Pokazuje tekst na ekranie. W logu długim zostanie zapisana bezwzględna pozycja i rozmiar bodźca w pikselach w formacie Left Top Width Height rozdzielone znakami tabulacji gdzie:  
Left – odległość od lewej krawędzi ekranu  
Top – odległość od góry ekranu  
Width – szerokość  
Height – wysokość

### 3.7.9 text\_hide

**text\_hide <x>** Ukrywa tekst.

### 3.7.10 text\_vpos

**text\_vpos <x> <pozycja>**  
Pozycjonowanie pionowe względne . Patrz 3.11.1

### 3.7.11 text\_hpos

**text\_hpos <x> <pozycja>**  
Pozycjonowanie poziome względne. Patrz 3.11.1

### 3.7.12 text\_vpospx

**text\_vpospx <x> <pozycja>**  
Pozycjonowanie pionowe bezwzględne. Patrz 3.11.2

### 3.7.13 text\_hpospx

**text\_hpospx <x> <pozycja>**  
Pozycjonowanie poziome bezwzględne. Patrz 3.11.2

Przykładowe wykorzystanie komend definiujących bodźce tekstowe.

```
; Wyświetlenie dużego napisu ZIELONY w kolorze czerwonym po prawej
stronie ekranu (w połowie pomiędzy środkiem a prawą krawędzią)

text_text 1 DZIĘKUJEMY, KONIEC BADANIA // Treść bodźca
text_text 1 DZIĘKUJEMY<br>KONIEC BADANIA // Treść jak wyżej,
// ale tekst będzie w dwóch liniach
text_text 1 DZIĘKUJEMY<br><br>KONIEC BADANIA
// W tym przypadku pomiędzy liniami będzie pusta linia .

text_color 1 red // kolor bodźca
text_size 1 75 // rozmiar bodźca = 75 punktów
text_hpos 1 50 // pozycja pozioma względna bodźca
```

## 3.8 Obrazy

Obrazki (pliki z grafikami) są drugą formą prezentowania treści osobie badanej. Można stosować do 999 predefiniowanych bodźców i dowolnie manipulować pozycją obrazka.

W poniższych komendach **<x>** oznacza numer obrazka – od 1 do 999

**UWAGA:** pliki powinny być w rozmiarze dopasowanym do monitora – program nie skaluje (zmniejsza, zwiększa) obrazków. Trzeba do tego zastosować program graficzny, np. FastStone

### 3.8.1 pic\_load

**pic\_load <x> <file>**

Importuje obrazek z pliku do pamięci komputera i zapamiętuje go jako numer **<x>**. Plik może być w formacie PNG, BPM, JPG, GIF.

Dla wygody można zdefiniować import i parametry obrazków na samym początku pliku BPX – skrypt będzie bardziej czytelny, a wgrywanie obrazków z wolnych nośników (pendrive) nie spowoduje powstania opóźnień.

Domyślnie obrazek jest wycelowany względem środka ekranu – jeżeli chcemy go przesunąć należy użyć komendy pozycjonujące.

### 3.8.2 pic\_name

**pic\_name <x> <tekst>**

Nadaje nazwę obiektowi. **<tekst>** jest dowolną nazwą - łańcuchem znaków w jednej linii. Jeżeli nadamy obiektowi nazwę to będzie ona używana w logu krótkim w komendzie **log\_input** i **timer\_input** zamiast identyfikatora.

Nazwy nie muszą być unikalne (mogą się powtarzać) – służą jedynie do wygodniejszego logowania zdarzeń

### 3.8.3 pic\_show

**pic\_show <x>**

Pokazuje obrazek na ekranie. W logu długim zostanie zapisana bezwzględna pozycja i rozmiar bodźca w pikselach w formacie Left Top Width Height rozdzielone znakami tabulacji gdzie:

Left	– odległość od lewej krawędzi ekranu
Top	– odległość od góry ekranu
Width	– szerokość
Height	– wysokość

### 3.8.4 pic\_hide

**pic\_hide <x>**

Ukrywa obrazek.

### 3.8.5 pic\_vpos

**pic\_vpos <x> <pozycja>**

Pozycjonowanie pionowe względne . Patrz 3.11.1

### 3.8.6 pic\_hpos

**pic\_hpos <x> <pozycja>**

Pozycjonowanie poziome względne. Patrz 3.11.1

### 3.8.7 pic\_vpospx

**pic\_vpospx <x> <pozycja>**

Pozycjonowanie pionowe bezwzględne. Patrz 3.11.2

### 3.8.8 pic\_hpospx

**pic\_hpospx <x> <pozycja>**

Pozycjonowanie poziome bezwzględne. Patrz 3.11.2

```

; Przykład wyświetlenia dwóch obrazków obok siebie i oczekiwanie
; na wybranie jednego z nich - kliknięcia w tło będą rejestrowane
; ale ignorowane

pic_load 1 puppy.jpg           // wgraj obraz nr 1 z pliku
pic_hpos 1 -50                 // ustaw obraz po lewej stronie ekranu
pic_load 2 jezyk.jpg           // wgraj obraz nr 2 z pliku
pic_hpos 2 50                  // ustaw obraz po prawej stronie ekranu
pic_show 1                     // wyświetl obraz nr 1
pic_show 2                     // wyświetl obraz nr 2
wait_mouse picture_001, picture_002 // czeka aż osoba badana kliknie
                                     // obrazek nr 1 lub obrazek nr 2

```

### 3.9 Odtwarzanie i nagrywanie dźwięków

#### 3.9.1 sound\_play

**sound\_play <file>**

odtwarza plik dźwiękowy w formacie WAV lub MP3. W obecnej wersji programu można odtwarzać jedynie jeden plik dźwiękowy na raz. Pliki dźwiękowe nie mogą być wgrane przed prezentacją do pamięci komputera – jeżeli będą bardzo duże a nośnik wolny (np. pendrive) mogą wprowadzić opóźnienie w rozpoczęciu dźwięku.

#### 3.9.2 sound\_stop

**sound\_play**

Zatrzymuje odtwarzanie plików dźwiękowych.

#### 3.9.3 beep

**beep**

Odtwarza krótki sygnał dźwiękowy – częstotliwość 4kHz i długość 100 ms.

#### 3.9.4 record\_start

**record\_start**

Rozpoczyna nagrywanie dźwięku przy pomocy mikrofonu komputera. Plik dźwiękowy zostanie automatycznie zapisany do katalogu programu BehaPresenter z nazwą taką samą jak plik BPS i z dodanymi kolejnymi numerami plików.

Nagrywanie dźwięków wymaga obecności pliku wzorcowego „BehaPresenter.wav” w katalogu programu BehaPresenter. Jeżeli komputer nie posiada mikrofonu powstanie plik pusty

#### 3.9.5 record\_stop

**record\_stop**

Zatrzymuje nagrywanie dźwięku i zapisuje plik na dysku.

```

; Wyświetlenie dużego napisu ZIELONY w kolorze czerwonym i
; nagranie 20 sekund dźwięku z mikrofonu po wyświetleniu

text_text 1 ZIELONY           // Treść bodźca
text_color 1 red               // kolor bodźca
text_size 1 75                // rozmiar bodźca = 75 punktów
record_start                   // rozpoczyna nagrywanie z mikrofonu
text_show 1                    // pokazuje bodziec na ekranie
wait_time 20000                // czeka 20 sekund
record_stop                    // zatrzymuje nagrywanie

```

## 3.10 Odtwarzanie filmów - plików wideo

Program ma możliwość stosowania jako bodźce filmy (pliki wideo). Jednocześnie może być załadowany i odtwarzany tylko jeden film. Filmy najlepiej konwertować do formatu AVI i ustawić w pliku doświadczalnym prawidłowy rozmiar filmu.

**UWAGA:** Typ kodowania pliku wideo należy przetestować na komputerze docelowym (na którym będzie się prowadzić doświadczenia). To czy film będzie prawidłowo odtwarzany zależy od systemu operacyjnego i zainstalowanych kodeków.

### 3.10.1 video\_load

**video\_load** <file>

Importuje film z pliku do pamięci komputera. Plik powinien być w formacie AVI.

Dla wygody można zdefiniować import i parametry na samym początku pliku BPX – skrypt będzie bardziej czytelny, a wgrywanie filmu z wolnych nośników (pendrive) nie spowoduje powstania opóźnień.

Domyślnie film jest wycelowany względem środka ekranu – jeżeli chcemy go przesunąć należy użyć komendy pozycjonujące.

### 3.10.2 video\_play

**video\_play** [<czas>]

Uruchamia odtwarzanie filmu. Opcjonalny parametr pozwala uruchomić plik wideo na określony czas podany w milisekundach.

### 3.10.3 video\_stop

**video\_stop**

Zatrzymuje odtwarzanie pliku.

### 3.10.4 video\_vpos

**video\_vpos** <pozycja>

Pozycjonowanie pionowe względne . Patrz 3.11.1

### 3.10.5 video\_hpos

**video\_hpos** <pozycja>

Pozycjonowanie poziome względne. Patrz 3.11.1

### 3.10.6 video\_vpoxpx

**video\_vpoxpx** <pozycja>

Pozycjonowanie pionowe bezwzględne. Patrz 3.11.2

### 3.10.7 video\_hpoxpx

**video\_hpoxpx** <pozycja>

Pozycjonowanie poziome bezwzględne. Patrz 3.11.2

### 3.10.8 video\_width

**video\_width** <szerokosc>

Ustawia szerokość okna filmu

### 3.10.9 video\_height

**video\_height** <wysokosc>

Ustawia wysokość okna filmu

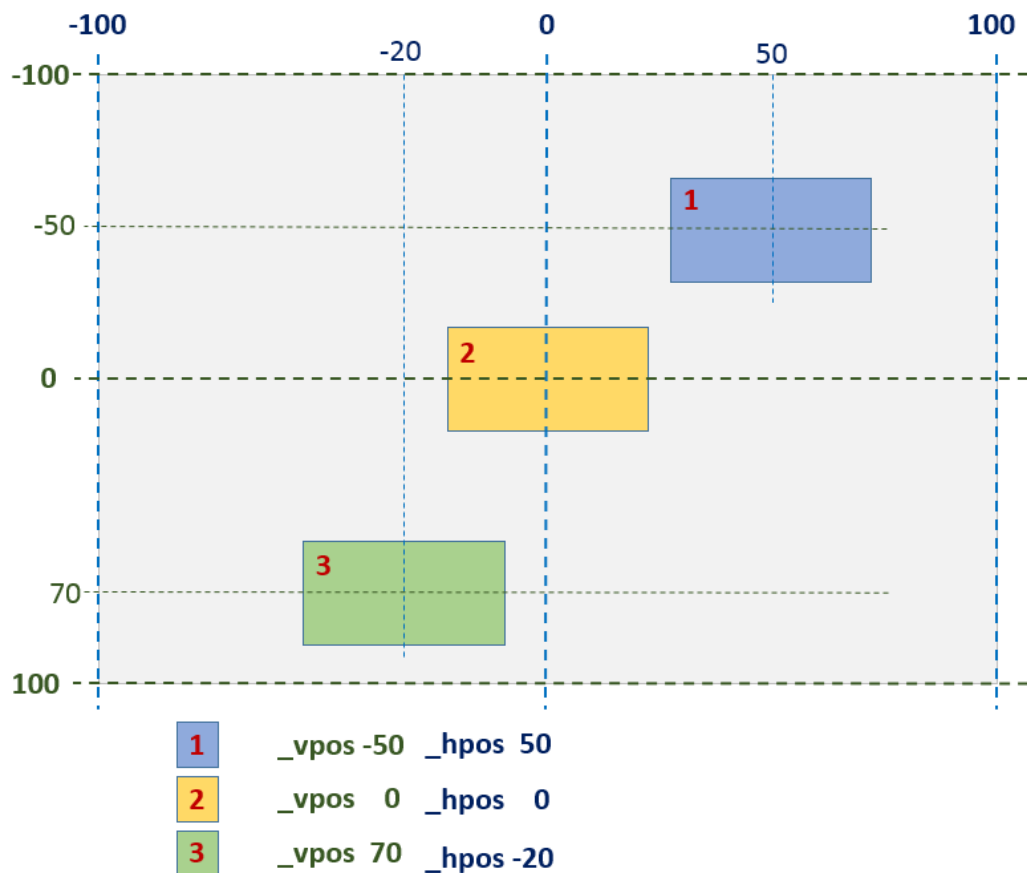


### 3.11 Pozycjonowanie obiektów na ekranie

Pozycje na ekranie możemy określić w sposób względny – procent przesunięcia względem środka ekranu – i bezwzględny – przesunięcie w pikselach względem lewego, górnego rogu. Pozycjonowanie względne pozwala na prezentację tego samego zestawu bodźców na monitorach o różnym rozmiarze i rozdzielczości. Pozycjonowanie bezwzględne pozwala na bardzo dokładne określenie pozycji.

#### 3.11.1 Pozycjonowanie względne

Pozycja względem środka ekranu w procentach. VPOS ujemne przesuwają obiekt do góry, dodatnie do dołu. HPOS ujemne przesuwają w lewo, dodatnie w prawo. Obiekt (obrazek, tekst) jest centrowany względem swojego środka



### 3.11.2 Pozycjonowanie bezwzględne

Pozycja względem górnego lewego rogu ekranu w pikselach – rozdzielczości ekranu. VPOSPX przesuwa obiekt do dołu. HPOSPX przesuwa obiekt w prawo. Obiekt (obrazek, tekst) jest centrowany względem swojego lewego górnego rogu.



## 3.12 Kolory

### 3.12.1 Definiowanie kolorów

Kolory możemy definiować na kilka sposobów:

- **nazwa koloru po angielsku** – lista nazw kolorów podana jest w dodatku 5.2 i również dostępna pod tym adresem:  
[http://docwiki.embarcadero.com/RADStudio/XE6/en/Colors\\_in\\_VCL](http://docwiki.embarcadero.com/RADStudio/XE6/en/Colors_in_VCL)
- **Liczba szesnastkowa w formacie HTML** – format **#RGB** (czerwony-zielony-niebieski) pozwala uzyskać dowolny kolor z palety. Można posłużyć się gotowymi paletami lub odczytać kolor z tła obrazków.  
[http://www.kurshtml.edu.pl/html/wykaz\\_kolorow,kolory.html](http://www.kurshtml.edu.pl/html/wykaz_kolorow,kolory.html)  
<http://fonty.pl/porady,15,kolory.htm>
- **Liczba szesnastkowa w formacie Delphi** – format **\$BGR** (niebieski-zielony-czerwony) lub **\$00BGR**, czyli tak jak to jest w tabeli w dodatku 5.2.

### 3.12.2 Dopasowanie koloru tła

Jeżeli chcemy dopasować tło prezentacji do tła obrazka to trzeba znaleźć kolor RGB - można to zrobić np. używając programu Paint w Windows – zaimportować obrazek, użyć pipetki i odczytać wartość RGB lub stosując programy odczytujące kolory z ekranu, np. program Color Seizer (patrz 5.1.3)

## 4 Pliki wynikowe - logi

### 4.1 Plik BPS

Po zakończeniu doświadczenia program zapisuje na dysku w katalogu z którego został uruchomiony plik **BPS** - log doświadczenia z nagłówkami.

Nazwa pliku jest generowana domyślnie i składa się z nazwy pliku eksperymentalnego + znacznik czasowy uruchomienia eksperymentu. Dzięki temu pliki wynikowe nie zostaną nadpisane na sobie.

Pierwsza część zawiera informacje o skrypcie i komputerze

Krótki log [LOG] zawiera jedynie informacje, które chcieliśmy uzyskać przy pomocy komend wymienionych w punkcie 3.5. Jeżeli nie zastosujemy żadnej z tych komend to log krótki będzie pusty.

Długi log [LOG LONG] zawiera wszystkie zdarzenia – definicje i prezentacje bodźców oraz aktywność osoby badanej.

Pierwsza kolumna zawiera czas w milisekundach liczony od rozpoczęcie działania skryptu / uruchomienia programu

Druga kolumna zawiera czas w milisekundach liczony od rozpoczęcia poprzedniej komendy – może nam pokazać czas wykonania prezentacji, opóźnienia etc.

Trzecia kolumna to komendy z pliku BPX wraz z odpowiadającymi im parametrami, np. dokładną informacją o położeniu bodźca. Są tam zapisane wszystkie aktywności osoby badanej – które klawisze i jak długo wciskała (rejestrowane są wszystkie klawisze, również równoległe wciskanie kilku klawiszy) oraz kliknięcia myszką – który klawisz i który obiekt został kliknięty.

```
experiment      :      experiment 001
group           :      Control
subject        :      HM
date            :      2015.05.05
time            :      20:37:17.363
sequ           :      ex_test_03.bpx
file            :      ex_test_03_20150505_203717.bps
soft            :      BehaPresenter
soft_ver        :      1.0.0
soft_date       :      2015.05.05
screen_no       :      0
screen_x        :      1920
screen_y        :      1080
screen_fr       :      60
compname        :      user-PC
compmac         :      xxxxxxxxxxxxx
username        :      user

=== [LOG] =====
text_001
Odpowiedz
text_001        12705
=== [LONG LOG] =====
    10      10 text_font 1 Arial
    17       7 text_color 1 green
    17       0 text_bgcolor 1 silver
    17       0 text_size 1 200
    23       6 text_text 1 TEST
    23       0 timer set 1
```



## 5 Dodatki

### 5.1 Przydatne programy

Proponowane narzędzia są programami darmowymi, których licencja pozwala na swobodne stosowanie również do celów komercyjnych. Są to programy tzw. portable – nie wymagają instalacji, więc nie ma problemów z zabezpieczeniami systemu operacyjnego – nie potrzebne jest konto administratora.

#### 5.1.1 FastStone Image Viewer



Darmowy program do przeglądania, edycji i konwersji plików graficznych. Pozwala w wygodny sposób konwertować i zmieniać obrazki zbiorowo – np. dopasowywać do rozmiaru, poprawiać kontrast etc. Może być instalowany, ale również działa bez instalacji  
<http://www.faststone.org/FSViewerDetail.htm>

#### 5.1.2 Notepad++



Darmowy edytor plików tekstowych z ogromnymi możliwościami – zaletą jest brak konieczności instalowania (wystarczy rozpakować i działa).  
<http://notepad-plus-plus.org/>

#### 5.1.3 Color Seizer



Darmowy program do odczytywania kolorów z ekranu komputera – pozwala na odczytanie wartości RGB i ustawienie koloru tła tak, by bodziec z pliku graficznego wydawał się być bez ramki. Nie trzeba go instalować, działa bezpośrednio po rozpakowaniu.  
<http://www.alexnolan.net/software/colorseizer.htm>

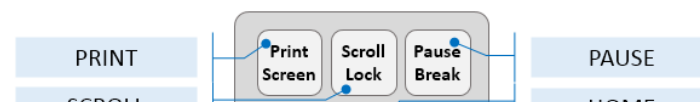
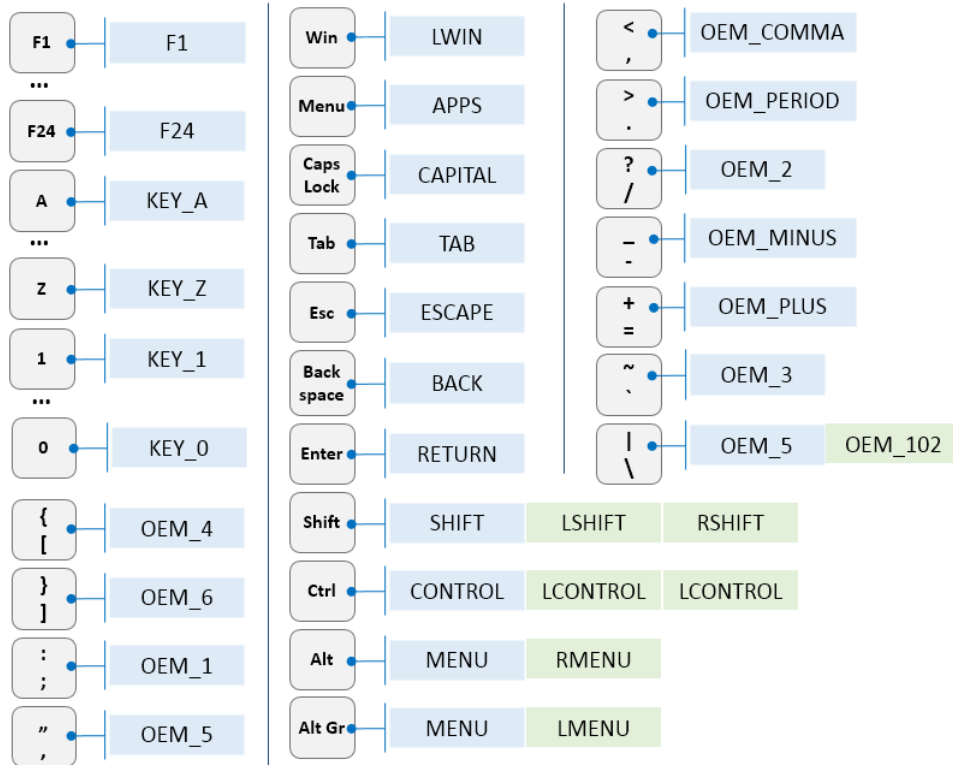
#### 5.1.4 Audacity



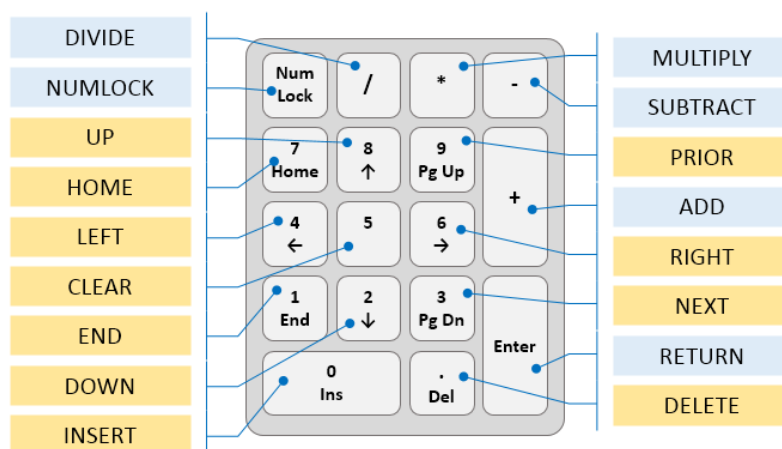
Darmowy program do edycji dźwięków – można go zastosować do przygotowania bodźców dźwiękowych, ich konwersji na format WAV i do odczytu i wizualnego zmierzenia czasów odpowiedzi ustnych osób badanych.  
<http://sourceforge.net/projects/audacity/>  
Jest też wersja nie wymagająca instalacji  
[http://portableapps.com/apps/music\\_video/audacity\\_portable](http://portableapps.com/apps/music_video/audacity_portable)

## 5.2 Kody klawiszy

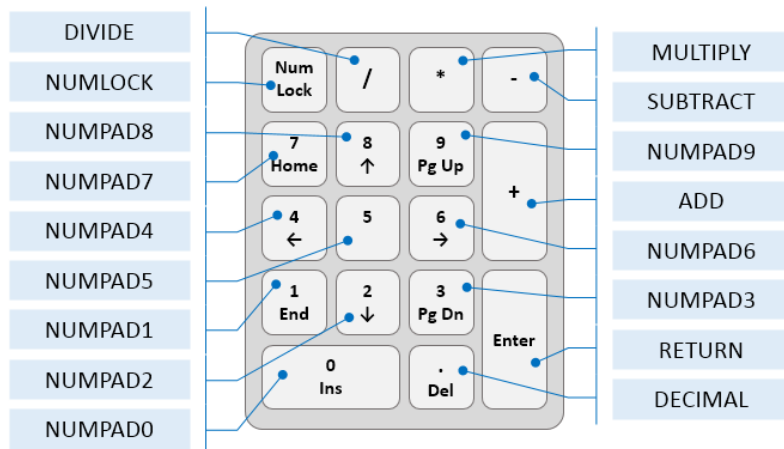
W komendach, które oczekują na wciśnięcie klawisza należy posługiwać się kodami (nazwami klawiszy). Nazwy są standardowe dla środowiska Windows, ale poszczególne klawiatury mogą różnić się pomiędzy sobą. Jeżeli mamy wątpliwości należy użyć narzędzia **KeyCodes.exe**



Num Lock OFF



## Num Lock ON





## 5.3 Tabela kolorów

Kolory: [http://docwiki.embarcadero.com/RADStudio/XE6/en/Colors\\_in\\_VCL](http://docwiki.embarcadero.com/RADStudio/XE6/en/Colors_in_VCL)

Value	Hex Value	Color			
None	\$1FFFFFFF			WebAzure	\$FFFFFF0
Aqua	\$FFFFFF00			WebMintcream	\$FAFF5
Black	\$000000			WebHoneydew	\$F0FF0
Blue	\$FF0000			WebLinen	\$E6F0FA
Cream	\$F0FBFF			WebLemonChiffon	\$CDFAFF
DkGray	\$808080			WebBlanchedAlmond	\$CDEBFF
Fuchsia	\$FF00FF			WebBisque	\$C4E4FF
Gray	\$808080			WebPeachPuff	\$B9DAFF
Green	\$008000			WebTan	\$8CB4D2
Lime	\$00FF00			WebYellow	\$00FFFF
LtGray	\$C0C0C0			WebDarkOrange	\$008CFF
Maroon	\$000080			WebRed	\$0000FF
MedGray	\$A4A0A0			WebDarkRed	\$00008B
MoneyGreen	\$C0DCC0			WebMaroon	\$000080
Navy	\$800000			WebIndianRed	\$5C5CCD
Olive	\$008080			WebSalmon	\$7280FA
Purple	\$800080			WebCoral	\$507FFF
Red	\$0000FF			WebGold	\$00D7FF
Silver	\$C0C0C0			WebTomato	\$4763FF
SkyBlue	\$F0CAA6			WebCrimson	\$3C14DC
Teal	\$808000			WebBrown	\$2A2AA5
White	\$FFFFFF			WebChocolate	\$1E69D2
Yellow	\$00FFFF			WebSandyBrown	\$60A4F4
WebSnow	\$FAFAFF			WebLightSalmon	\$7AA0FF
WebFloralWhite	\$F0FAFF			WebLightCoral	\$8080F0
WebLavenderBlush	\$F5F0FF			WebOrange	\$00A5FF
WebOldLace	\$E6F5FD			WebOrangeRed	\$0045FF
WebIvory	\$F0FFFF			WebFirebrick	\$2222B2
WebCornSilk	\$DCF8FF			WebSaddleBrown	\$13458B
WebBeige	\$DCF5F5			WebSienna	\$2D52A0
WebAntiqueWhite	\$D7EBFA			WebPeru	\$3F85CD
WebWheat	\$B3DEF5			WebDarkSalmon	\$7A96E9
WebAliceBlue	\$FFF8F0			WebRosyBrown	\$8F8FBC
WebGhostWhite	\$FFF8F8			WebPaleGoldenrod	\$AAE8EE
WebLavender	\$FAE6E6			WebLightGoldenrodYellow	\$D2FAFA
WebSeashell	\$EEF5FF			WebOlive	\$008080
WebLightYellow	\$E0FFFF			WebForestGreen	\$228B22
WebPapayaWhip	\$D5EFFF			WebGreenYellow	\$2FFAD
WebNavajoWhite	\$ADDEFF			WebChartreuse	\$00FF7F
WebMoccasin	\$B5E4FF			WebLightGreen	\$90EE90
WebBurlywood	\$87B8DE			WebAquamarine	\$D4FF7F
				WebSeaGreen	\$578B2E

WebGoldenRod	\$20A5DA	
WebKhaki	\$8CE6F0	
WebOliveDrab	\$238E6B	
WebGreen	\$008000	
WebYellowGreen	\$32CD9A	
WebLawnGreen	\$00FC7C	
WebPaleGreen	\$98FB98	
WebMediumAquamarine	\$AACD66	
WebMediumSeaGreen	\$71B33C	
WebDarkGoldenRod	\$0B86B8	
WebDarkKhaki	\$6BB7BD	
WebDarkOliveGreen	\$2F6B55	
WebDarkgreen	\$006400	
WebLimeGreen	\$32CD32	
WebLime	\$00FF00	
WebSpringGreen	\$7FFF00	
WebMediumSpringGreen	\$9AFA00	
WebDarkSeaGreen	\$8FBC8F	
WebLightSeaGreen	\$AAB220	
WebPaleTurquoise	\$EEEEAF	
WebLightCyan	\$FFFFE0	
WebLightBlue	\$E6D8AD	
WebLightSkyBlue	\$FACE87	
WebCornFlowerBlue	\$ED9564	
WebDarkBlue	\$8B0000	
WebIndigo	\$82004B	
WebMediumTurquoise	\$CCD148	
WebTurquoise	\$D0E040	
WebCyan	\$FFFF00	
WebAqua	\$FFFF00	
WebPowderBlue	\$E6E0B0	
WebSkyBlue	\$EBCE87	
WebRoyalBlue	\$E16941	
WebMediumBlue	\$CD0000	
WebMidnightBlue	\$701919	
WebDarkTurquoise	\$D1CE00	
WebCadetBlue	\$A09E5F	
WebDarkCyan	\$8B8B00	
WebTeal	\$808000	
WebDeepskyBlue	\$FFBF00	
WebDodgerBlue	\$FF901E	
WebBlue	\$FF0000	
WebNavy	\$800000	
WebDarkViolet	\$D30094	
WebDarkOrchid	\$CC3299	

WebMagenta	\$FF00FF	
WebFuchsia	\$FF00FF	
WebDarkMagenta	\$8B008B	
WebMediumVioletRed	\$8515C7	
WebPaleVioletRed	\$9370DB	
WebBlueViolet	\$E22B8A	
WebMediumOrchid	\$D355BA	
WebMediumPurple	\$DB7093	
WebPurple	\$800080	
WebDeepPink	\$9314FF	
WebLightPink	\$C1B6FF	
WebViolet	\$EE82EE	
WebOrchid	\$D670DA	
WebPlum	\$DDA0DD	
WebThistle	\$D8BFD8	
WebHotPink	\$B469FF	
WebPink	\$CBC0FF	
WebLightSteelBlue	\$DEC4B0	
WebMediumSlateBlue	\$EE687B	
WebLightSlateGray	\$998877	
WebWhite	\$FFFFFF	
WebLightgrey	\$D3D3D3	
WebGray	\$808080	
WebSteelBlue	\$B48246	
WebSlateBlue	\$CD5A6A	
WebSlateGray	\$908070	
WebWhiteSmoke	\$F5F5F5	
WebSilver	\$C0C0C0	
WebDimGray	\$696969	
WebMistyRose	\$E1E4FF	
WebDarkSlateBlue	\$8B3D48	
WebDarkSlategray	\$4F4F2F	
WebGainsboro	\$DCDCDC	
WebDarkGray	\$A9A9A9	
WebBlack	\$000000	