

# Screen Builder

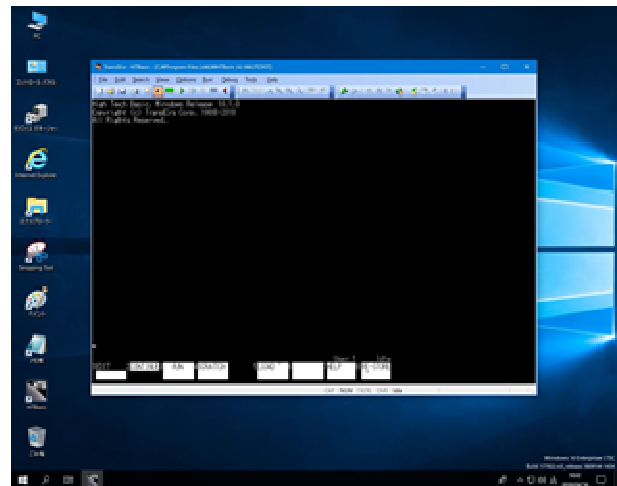
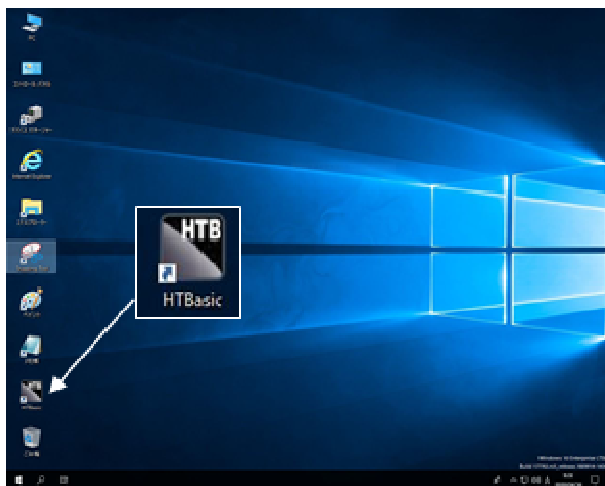
2020.04.16 更新（使用 PC : Core i7 Windows 10 Enterprise LTSC 180914 64bit 版）

2015.7 HTBasic 10.0.3 のリリースから、一部不具合のあった HTBasic BPLUS ウェジイトでの漢字の使用が修正されました。 またこのリリースから、BPLUS Screen Builder がリリースされました。

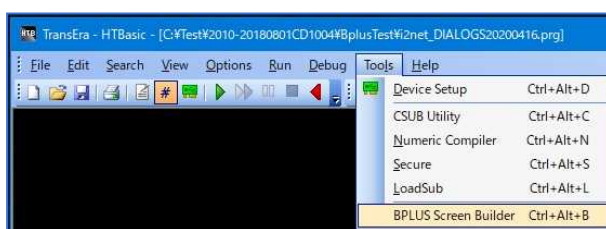
この資料は、BPLUS Screen Builder の補足資料としてアイネット株式会社が提供するものです。

## Screen Builder の起動

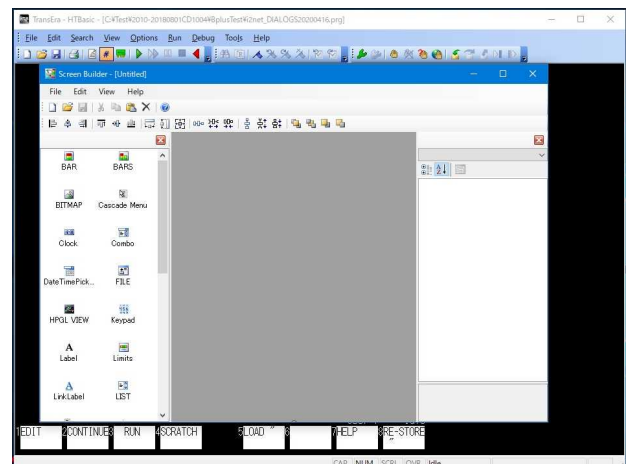
デスクトップの HTBasic 開発バージョンのショートカットアイコンから、HTBasic を起動、その後メニューバーから Tools | BPlus Screen Builder から開始します。



HTBasic の起動



Screen Builder の開始

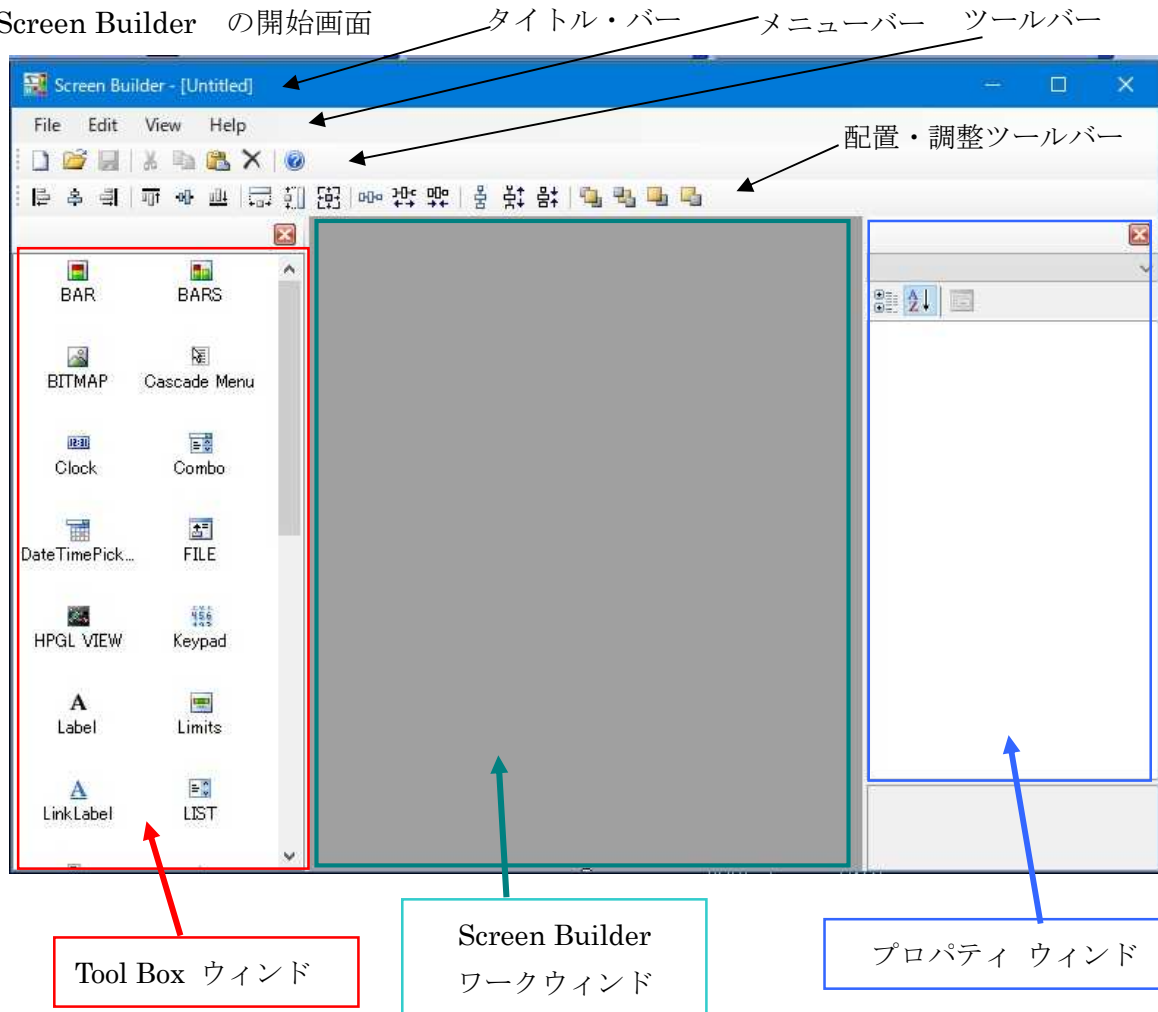


# Screen Builder

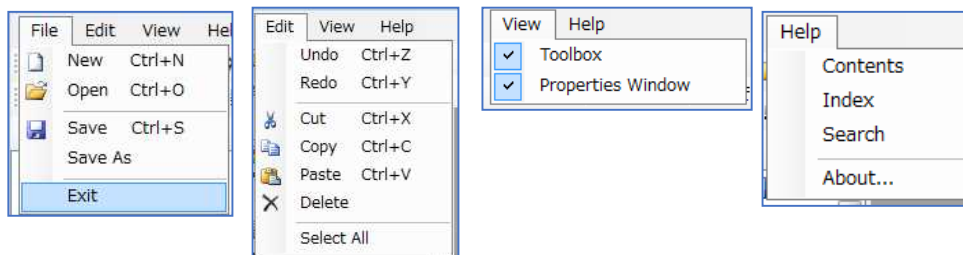
## Screen Builder の画面

1. Screen Builder の開始画面
2. メニューバー
3. メニューバーとツールバー
4. 配置・調整ツールバー
5. Tool Box のウィジェット

### 1. Screen Builder の開始画面






### 2 メニューバー



# Screen Builder

## 3. メニューバーとツールバー [ File ]

ツールバー	メニューバー	説明
	File   New	現在の SCRB のワーク・ウィンドウを閉じ、新しいワーク・ウィンドウを開く (ブランク)
	File   Open	保存されている SCRB ファイルを開く
	File   Save	現在開いている SCRB ファイルへの上書き保存 (変更がされていない状態では操作不能)
	File   Save As	現在作業中の、SCRB のワーク・ウィンドウをファイル名をつけて保存
	Exit	Screen Builder を終了

## [ Edit ]

ツールバー	メニューバー	説明
	Edit   Undo	一操作前に戻す
	Edit   Redo	操作を戻したところから、再操作
	Edit   Cut	現在選択されているウィジェットを削除(Cut)
	Edit   Copy	現在選択されているウィジェットをコピー(Copy)
	Edit   Paste	現在選択されているコピーしたウィジェットを PANEL に、貼り付け(Paste)
	Edit   Delete	現在選択されているウィジェットを削除(Delete)
	Select All	すべてを選択----->

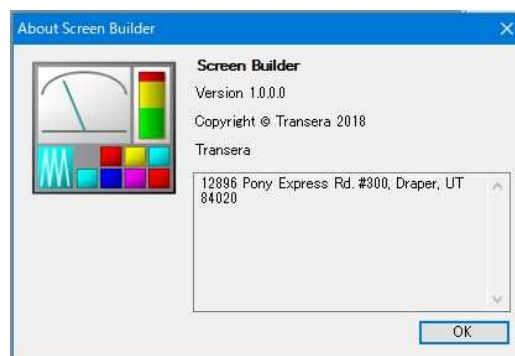
## [ View ]

ツールバー	メニューバー	説明
<input checked="" type="checkbox"/> Toolbox	View   Toolbox	ON : ToolBox Window の表示 OFF : ToolBox Window の非表示
<input checked="" type="checkbox"/> Properties Window	View   Properties Window	ON : Properties Window の表示 OFF : Properties Window の非表示

# Screen Builder

## [ Help ]

ツールバー	メニューバー	説明
	Help   Contents	HTBasic 10.1 のオンライン・ヘルプの実行目次タブをエントリーページとして表示
	Help   Index	HTBasic 10.1 のオンライン・ヘルプの実行キーワードタブをエントリーページとして表示
	Help   Serch	HTBasic 10. 1 のオンライン・ヘルプの実行検索タブをエントリーページとして表示
	Help   About	Screen Builder のバージョンを表示



Screen Builder のバージョンを表示 (1.0.0.0)

## 4. 配置・調整ツールバー

### 4-1 選択したウィジェットの配置あわせの調整 (横)




配置調整ツール	説明
	選択したウィジェットを左あわせする
	選択したウィジェットを中央あわせにする
	選択したウィジェットを右あわせする

### 4-2 選択したウィジェットの配置あわせの調整 (縦)




配置調整ツール	説明
	選択したウィジェットを上あわせにする
	選択したウィジェットを中央あわせにする
	選択したウィジェットを下あわせにする

## Screen Builder




### 4-3 選択したウィジェットのサイズあわせ

配置調整ツール	説 明
	選択したウィジェットの幅のサイズあわせ
	選択したウィジェットの高さのサイズあわせ
	選択したウィジェットの幅と高さのサイズあわせ





### 4-4 選択したウィジェットの間隔調整（横）

配置調整ツール	説 明
	選択したウィジェット間の横スペースを0にする
	選択したウィジェット間の横スペースを大きくする
	選択したウィジェット間の横スペースを小さくする (重なるようにもできます)

### 4-5 選択したウィジェットの間隔調整（縦）























配置調整ツール	説 明
	選択したウィジェット間の縦スペースを0にする
	選択したウィジェット間の縦スペースを大きくする
	選択したウィジェットの間の縦スペースを小さくする (重なるようにもできます)

### 4-6 選択したウィジェットの順序の変更





配置調整ツール	説 明
	選択したウィジェットを最前面に配置する
	選択したウィジェットを最背面に配置する
	選択したウィジェットを前面に配置する
	選択したウィジェットを背面に配置する



## Screen Builder

### 5. Tool Box のウィジェット

Tool Box	ウィジェット	Tool Box	ウィジェット
文字の表示			
 Label	LABEL	 LIST	LIST
 Printer	PRINTER		
データ出力			
 BAR	BAR	 BARS	BARS
 BITMAP	BITMAP	 Clock	CLOCK
 HPGL VIEW	HPGL VIEW	 Limits	LIMITS
 Meter	METER	 StripChart	STRIPCHART
 XY GRAPH	XY GRAPH		
データ入力			
 Combo	COMBO	 FILE	FILE
 Keypad	KEYPAD	 Number	NUMBER
 PushButton	PUSHBUTTON	 RadioButton	RADIOBUTTON
 ScrollBar	SCROLLBAR	 Slider	SLIDER
 String	STRING	 ToggleButton	TOGGLEBUTTON

# Screen Builder

メニューの生成			
 Cascade Menu	CASCADE MENU	 Menu Button	MENU BUTTON
 Menu Separator	MENU SEPARATOR	 Menu Toggle	MENU TOGGLE
 PullDown Menu	PULDOWN MENU		

Tool Box	ウィジェット	Tool Box	ウィジェット
特殊			
 Panel	PANEL	 Separator	SEPARATOR

## Screen Builder

---

### Screen Builder の 使い方・ 操作方法

例題) Screen Builder 使用して、.SRC ファイルを作成し、HTBasic のプログラムで実行する過程で使用方法・操作方法を理解するものです。

この例では、Screen Builder と SYSTEM ウィジェットを使用します。  
10本のワイヤーのうち、4本が接続/2本が爆発への引き金/4本は何も起こらないように配線され、いずれのワイヤーはどのようなになっているかを伏せて行う「Bomb Squad」という名前のゲームとして作られています。

#### [ルール]

- 4本のワイヤーを正しく切断できれば、爆弾は爆発から解除されます。
- もし間違った2本のワイヤーを切断すると、爆弾は爆発します
- 他の4本のワイヤーは何も行いませんので、切断しても何も起こりません

さらに、爆弾は爆発から解除させようと、「カッチ、カッチ」いっています。もし時間までに、爆発の解除ができないと、爆弾は爆発します

このゲームには、10個の TOGGLE BUTTON ウィジェット(ワイヤー)、TIMAR モードの CLOCK ウィジェット (爆発へのタイマー) と PRINTER ウィジェット(結果の表示)が、組み込まれます。ユーザー・インターフェースをデザインするために、Screen Builder アプリケーションを使っています。HTBasic10.0.3での完成例は、“BSQUAD3.SCR”を、実行プログラム “BOMB\_SQUAD\_LOAD” から読み込み実行します。

#### メモ

---

\*LOAD を使用してファイルからユーザー・インターフェースを読み込むところからこの例題は始まります。ウィジェットのイベントが設定され、このゲームのアルゴリズムが、TASK として、実行されます。この事については、オンライン・ヘルプにある “Bomb Squad (\*LOAD)” を参照してください。

---

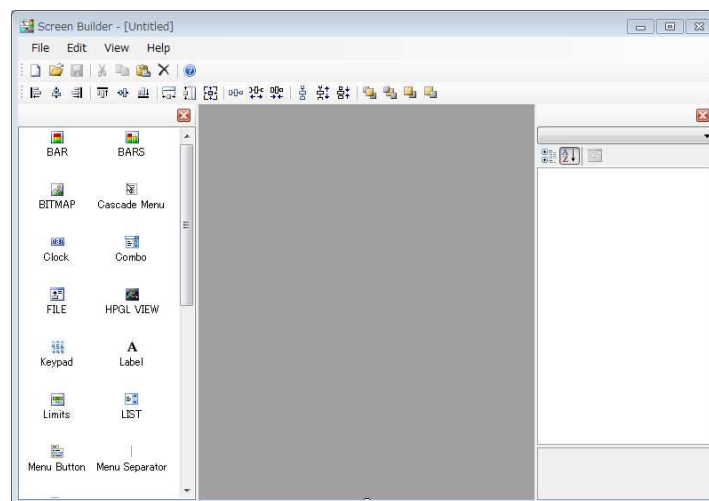


# Screen Builder

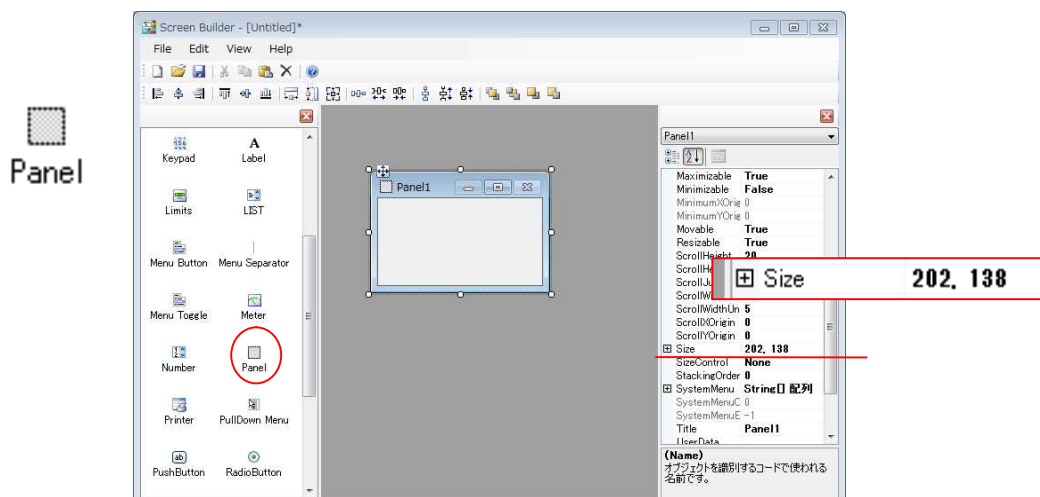
## Screen Builder の使い方・操作方法

Screen Builder 使用して、.SRC ファイルを作成するステップ

1. 前提 : HTBasic のプログラムは、HTbasic10 Install フォルダにある  
..¥Examples¥BPlus Examples¥BOMB\_SQUAD\_LOAD を使用  
30 LOAD BIN "BPLUS" ..... 30 行目を変更・追加  
430 CONTROL @Sys;SET ("\*LOAD":"BSQUAD3.SCR") ..... 430 行目を  
変更・追加  
変更後、BOMB\_SQUAD\_LOAD3.prg という名前で STORE してください。
2. Screen Builder で BSQUAD3.SCR を生成  
Step1 : メニューバーから、Tools | BPLUS Screen Builder で、Screen Builder  
アプリケーションを実行します。



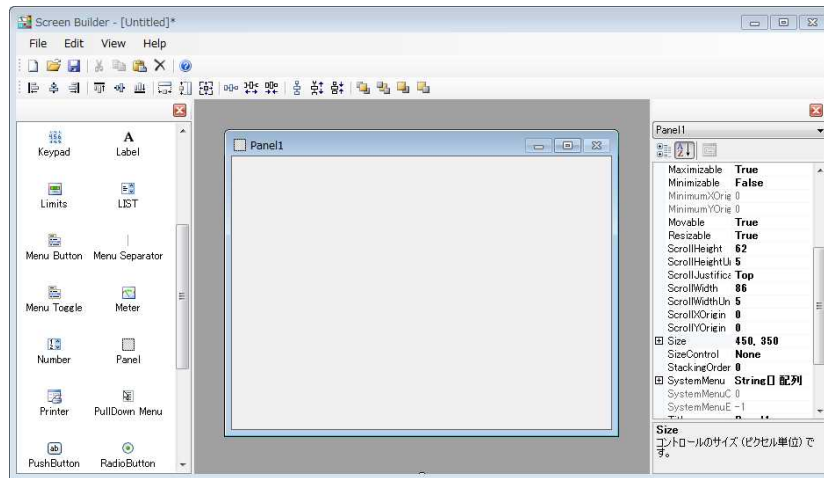
Step2 : ToolBox からパネル(PANEL)を選択、作業ウィンドにドラッグします。



Properties ウィンドの Size を 450,350 にし、全体が表示できるように  
Screen Builder Window の大きさを調整してください。

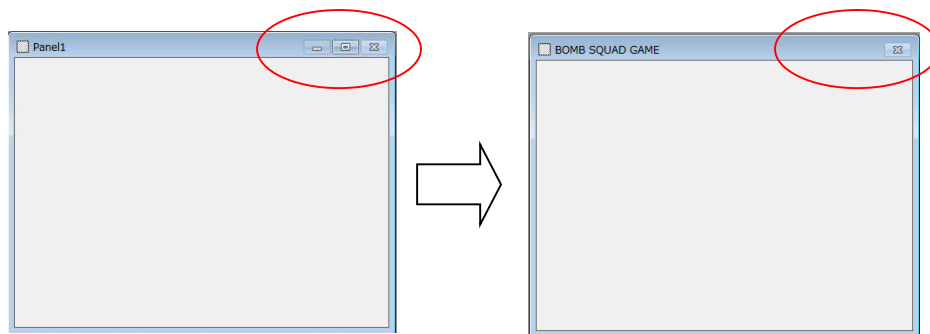
# Screen Builder

## Step3 : Panel1 のプロパティを変更

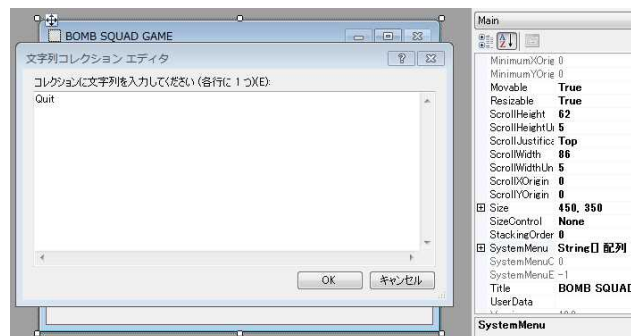


- ①パネル名 Panel1 を Main に変更
- ②TITLE を"BOMB SQUAD GAME"
- ③MAXIMIZEABLE と RESIZABLE を False に
- ④SYSTEM MENU に文字 "Quit"を

③



④

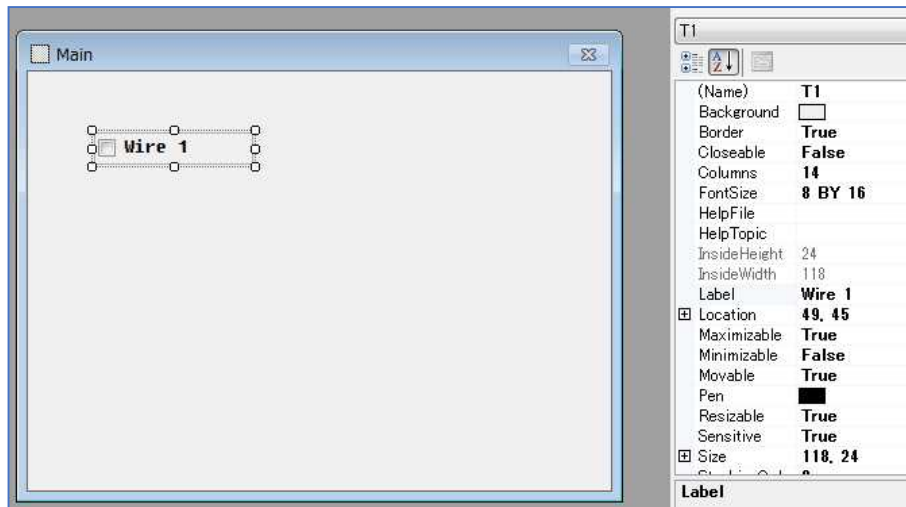


注) ②については変更後もブランクになりますが、表示状態で保存すれば正常に表示されます。(2015.07 時点の不具合)

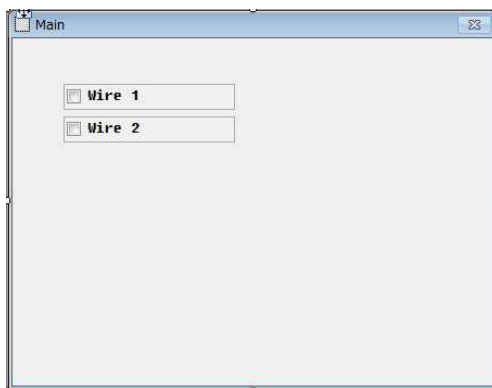
## Step4 : 10 個のトグルボタン(TOGGLEBUTTON)を作ります

- (1) ToolBox から Toggle Button(TOGGLEBUTTON)を選択、Main パネルに最初のボタンとしてドラッグします。 ウィジェット名を「T1」にし、Label を"Wire 1"とします。


# Screen Builder

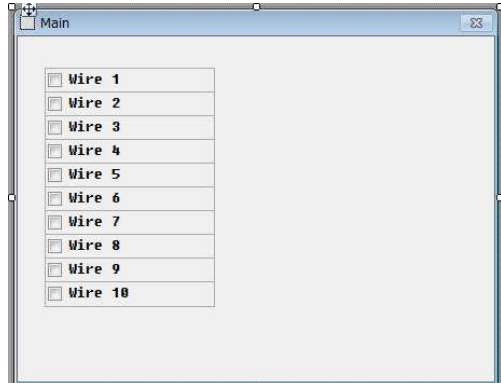
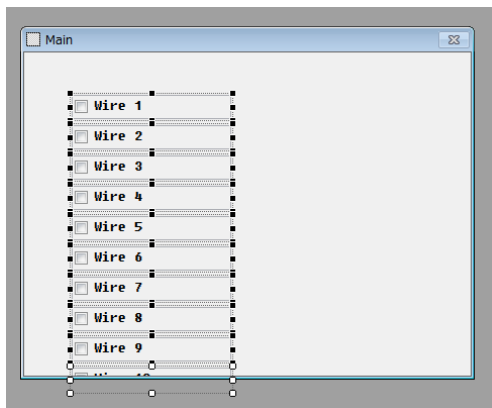


- (2) 残りの 9 個トグルボタン(TOGGLEBUTTON)をコピーを使用して、作ります。  
T1 を選択し、ツールバー  (Edit | Copy)  (Edit | Copy) の順にクリックし、T2 の取込をします。



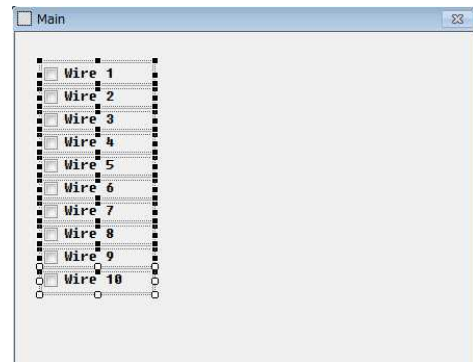
同様な作業を、10 個まで繰り返します。

- (3) ツールバーの  使用して選択したウィジェットの間の縦スペースを小さくし、その位置を変更します。

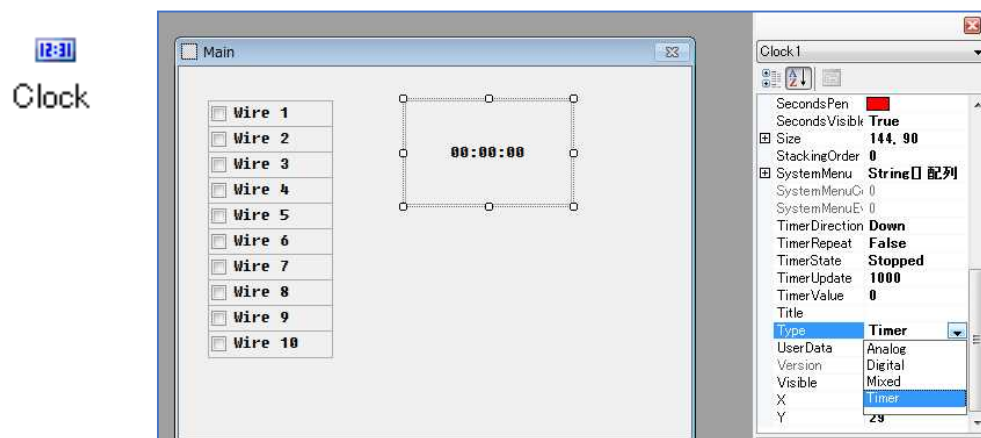


# Screen Builder

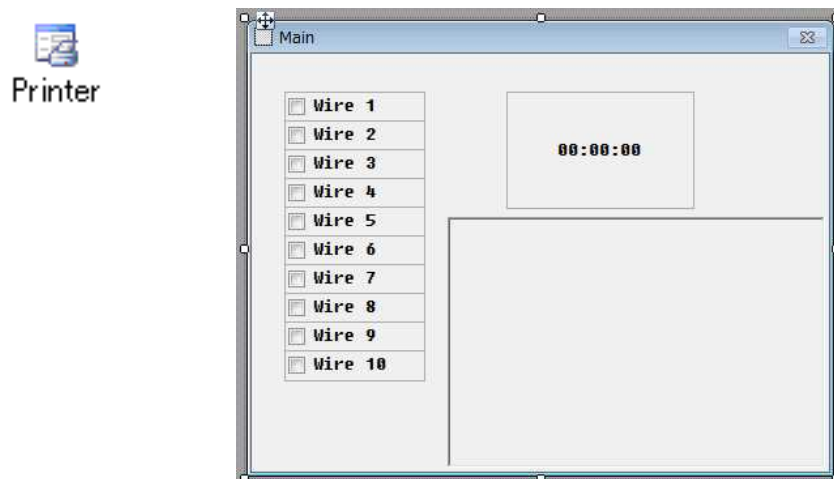
(4) 幅も 10 個選択し、一度に変更します。



Step5 : ToolBox から時計(CLOCK)を選択、Main パネルにドラッグします。  
ウィジェット名を「Clock」にし、TYPE を"TIMER"にします。

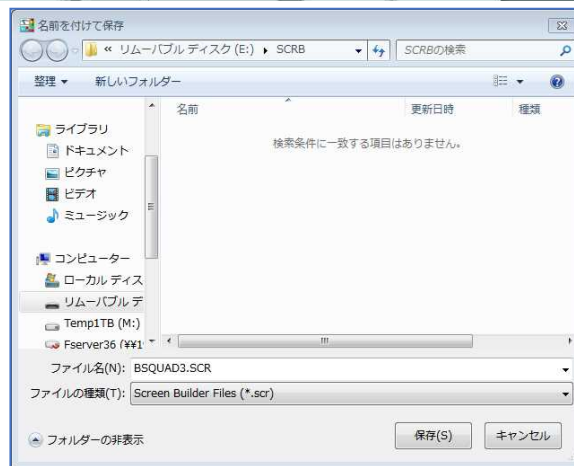
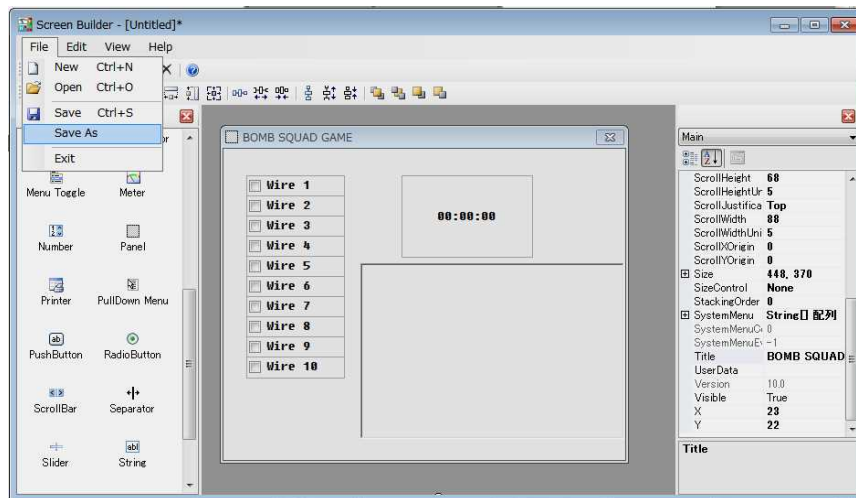


Step6 : ToolBox から Printer(PRINTER)を選択、Main パネルにドラッグします。  
ウィジェット名を「Printer」にします。 Text を、Null にします。



# Screen Builder

Step7 : Main パネルに戻り、TITLE を"BOMB SQUAD GAME"に再度設定し、「BSQUAD3.SCR」という名前で保存します。

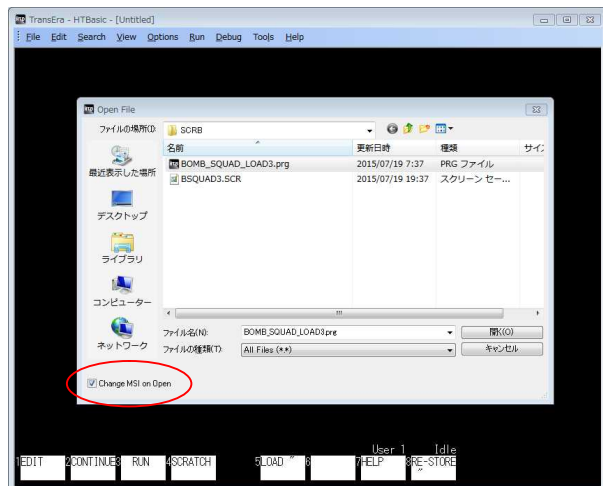


保存

# Screen Builder

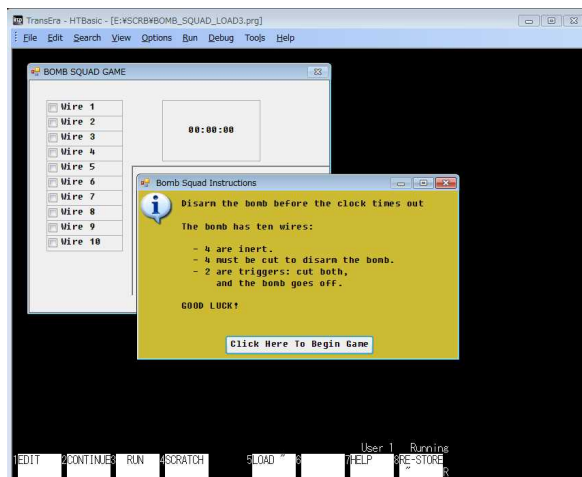
## 3. プログラム BOMB\_SQUAD\_LOAD3.prg を HTBasic で読み込み実行 (Change MSI On Open にチェック)

- (1) File | Open  
BOMB\_SQUAD\_LOAD3.prg  
を読み込みます。

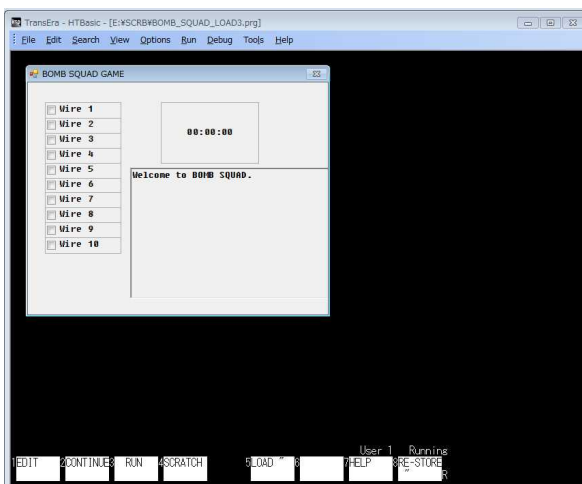


- (2) Run | Run  
プログラムを実行します。  
このプログラムが実行されると  
430 CONTROL @Sys;SET ("\*LOAD":"BSQUAD3.SCR")で  
BSQUAD3.SCR が読み込まれます。

このゲームの実行上の説明が  
出ますので、実行します



- (3) ゲーム実行中



以上

# Screen Builder

## Screen Builder 追加情報

例題) Screen Builder 使用して、.SRC ファイルを作成し、HTBasic のプログラムで実行しする過程で使用方法・操作方法を理解する。

### 1. 「BSQUAD3.SCR」 のファイル内容

<pre> 13 WIDGET 5:PANEL 4:Main -1 26 47 448 370 1 1:X 23 1:Y 22 11:MAXIMIZABLE 0 9:RESIZABLE 0 5:TITLE 15:BOMB SQUAD GAME 4::END  WIDGET 12:TOGGLEBUTTON 2:T1 0 52 77 108 24 1 13:STACKINGORDER 11 1:X 25 1:Y 29 7:COLUMNS 13 5:LABEL 6:Wire 1 4::END  WIDGET 12:TOGGLEBUTTON 2:T2 0 52 99 108 24 1 13:STACKINGORDER 10 1:X 25 1:Y 51 7:COLUMNS 13 5:LABEL 6:Wire 2 4::END  WIDGET 12:TOGGLEBUTTON 2:T3 0 52 121 108 24 1 13:STACKINGORDER 9 1:X 25 1:Y 73 7:COLUMNS 13 5:LABEL 6:Wire 3 4::END  WIDGET 12:TOGGLEBUTTON 2:T4 0 52 143 108 24 1 13:STACKINGORDER 8 1:X 25 1:Y 95 7:COLUMNS 13 5:LABEL 6:Wire 4 4::END  WIDGET 12:TOGGLEBUTTON 2:T5 0 52 165 108 24 1 13:STACKINGORDER 7 1:X 25 1:Y 117 7:COLUMNS 13 5:LABEL 6:Wire 5 4::END  WIDGET 12:TOGGLEBUTTON 2:T6 0 52 187 108 24 1 13:STACKINGORDER 6 1:X 25 </pre>	<pre> 1:Y 139 7:COLUMNS 13 5:LABEL 6:Wire 6 4::END  WIDGET 12:TOGGLEBUTTON 2:T7 0 52 209 108 24 1 13:STACKINGORDER 5 1:X 25 1:Y 161 7:COLUMNS 13 5:LABEL 6:Wire 7 4::END  WIDGET 12:TOGGLEBUTTON 2:T8 0 52 231 108 24 1 13:STACKINGORDER 4 1:X 25 1:Y 183 7:COLUMNS 13 5:LABEL 6:Wire 8 4::END  WIDGET 12:TOGGLEBUTTON 2:T9 0 52 253 108 24 1 13:STACKINGORDER 3 1:X 25 1:Y 205 7:COLUMNS 13 5:LABEL 6:Wire 9 4::END  WIDGET 12:TOGGLEBUTTON 3:T10 0 52 275 108 24 1 13:STACKINGORDER 2 1:X 25 1:Y 227 7:COLUMNS 13 5:LABEL 7:Wire 10 4::END  WIDGET 5:CLOCK 6:Clock1 0 222 77 144 90 1 13:STACKINGORDER 1 1:X 195 1:Y 29 4:TYPE 3 4::END  WIDGET 7:PRINTER 7:Printer 0 179 175 289 192 1 1:X 150 1:Y 125 7:COLUMNS 36 11:HIDDENLINES 32756 4:ROWS 11 4:TEXT 0: 4::END </pre>
--	---



# Screen Builder

## 2. BOMB\_SQUAD\_LOAD3.prg の内容

<pre> 10      ! ***** ***** 20      ! Example: Bomb Squad (*LOAD Version) 30 LOAD BIN "BPLUS" 40      ! This program demonstrates the use of the       CLOCK widget in TIMER mode 50      ! by playing a game in which the user has to       disarm a bomb by cutting 60      ! wires. There are 10 wires. You must cut the       correct four to disarm 70      ! the bomb. Of the six wires left, four are       don't-cares and two cause 80      ! the bomb go off immediately. 90      ! 100     ! The wires are represented by 10       TOGGLEBUTTONs. This program uses the 110     ! SYSTEM widget to create the       TOGGLEBUTTONs. This is convenient, since 120     ! otherwise you would have to have some       degree of separate code for 130     ! each TOGGLEBUTTON. The fact that you       only get one event for all 140     ! 10 TOGGLEBUTTONs is OK, too, since       each time a TOGGLEBUTTON event 150     ! happens the program scans through all the       TOGGLEBUTTONs to check 160     ! their VALUEs. 170     ! 180     ! ***** ***** 190     ! 200     RANDOMIZE       INT(10^7*FRACT(TIMEDATE))! Set random seed 210     ! 220     ! Miscellaneous general-purpose variables 230     ! 240     INTEGER N,V 250     DIM S\$[256],B\$(1:1)[64],Eol\$[2] 260     Eol\$=CHR\$(13) 270     ! 280     ! Variables for playing the game 290     ! 300     INTEGER Playgame      ! Indicates       game in progress 310     INTEGER Wires(1:10)  !       Designates wire settings 320     INTEGER Live,Kill,Dontcare,Cut! Wire       values 330     DATA 1,2,3,4 340     READ Live,Deadly,Dontcare,Cut 350     INTEGER Livewires,Lethal! Number of       live/deadly wires 360     ! 370     CLEAR SCREEN 380     ! 390     ! Create a SYSTEM widget and load Screen       Builder file 400     ! 410     COM @Sys 420     ASSIGN @Sys TO WIDGET "SYSTEM" 430     CONTROL @Sys:SET       (**LOAD:"BSQUAD3.SCR") </pre>	<pre> 440     ! 450     ! Set up events for SYSTEM MENU,       TOGGLEBUTTONs, and CLOCK 460     ! 470     ON EVENT @Sys,"SYSTEM MENU",15       GOTO Finis 480     ON EVENT @Sys,"CHANGED" GOSUB       Cutwire 490     ON EVENT @Sys,"TIMER" GOSUB       Boomboom 500     ! 510     ! Turn on panel 520     ! 530     CONTROL @Sys:SET       (**NAME:"Main","VISIBLE":1) 540     ! 550     ! Display instructions using a DIALOG 560     ! 570     S\$="Disarm the bomb before the clock times       out"&amp;Eol\$ 580     S\$=S\$&amp;Eol\$ 590     S\$=S\$&amp;"The bomb has ten wires:"&amp;Eol\$ 600     S\$=S\$&amp;Eol\$ 610     S\$=S\$&amp;" - 4 are inert."&amp;Eol\$ 620     S\$=S\$&amp;" - 4 must be cut to disarm the       bomb."&amp;Eol\$ 630     S\$=S\$&amp;" - 2 are triggers: cut both, "&amp;Eol\$ 640     S\$=S\$&amp;" and the bomb goes off."&amp;Eol\$ 650     S\$=S\$&amp;Eol\$ 660     S\$=S\$&amp;"GOOD LUCK!"&amp;Eol\$ 670     ! 680     B\$(1)="Click Here To Begin Game" 690     ! 700     DIALOG "INFORMATION",S\$:SET       (**TITLE:" Bomb Squad       Instructions","BACKGROUND":9,"PEN":0,"JUSTIFICA       TION":"LEFT","DIALOG BUTTONS":B\$(*)) 710     ! 720     ! Main game loop. 730     ! 740     LOOP 750     ! 760     ! Clear PRINTER widget, set up all the "wires". 770     ! 780     DISABLE 790     CONTROL @Sys:SET       (**NAME:"Main/Printer","TEXT":") 800     CALL Pr("Welcome to BOMB SQUAD.") 810     CALL Pr("") 820     ! 830     FOR N=1 TO 10 840         Wires(N)=Dontcare 850         S\$=VAL\$(N) 860         CONTROL @Sys:SET       (**NAME:"Main/T"&amp;S\$) 870         CONTROL @Sys:SET       (**SENSITIVE":1,"VALUE":0) 880     NEXT N 890     ! 900     ! Set the deadly wires. 910     ! 920     Lethal=0 </pre>
---	---



# Screen Builder

930	REPEAT	1490	!
940	N=1+INT(10*RND)	1500	! Ignore the button if it is not set, otherwise ...
950	IF Wires(N)=Dontcare THEN	1510	!
960	Wires(N)=Deadly	1520	IF V=1 THEN
970	Lethal=Lethal+1	1530	!
980	END IF	1540	! ... disable the button ...
990	UNTIL (Lethal=2)	1550	!
1000	!	1560	CONTROL @Sys;SET
1010	! Set up the live wires.	1570	("SENSITIVE":0)
1020	!	1580	! ... and take the appropriate measures for the
1030	Livewires=0	1590	wire value.
1040	REPEAT	1600	SELECT Wires(N)
1050	N=1+INT(10*RND)	1610	!
1060	IF Wires(N)=Dontcare THEN	1620	! Don't care, just say so.
1070	Wires(N)=Live	1630	!
1080	Livewires=Livewires+1	1640	CASE Dontcare
1090	END IF	1650	CALL Pr("Inert wire.")
1100	UNTIL (Livewires=4)	1660	!
1110	!	1670	! Live wire: decrement the live-wire count -- if it
1120	! Set the timer to 30 seconds, start it running.	1680	reaches 0,
1130	!	1690	! you win. Use a DIALOG to indicate the matter
1140	CONTROL @Sys;SET	1700	and query to see
1150	CONTROL @Sys;SET ("TIMER	1710	! if the user wants to play another game. If it
1160	VALUE":30000,"TIMER STATE":	1720	is not 0,
1170	! Loop until game over. Note how the	1730	! announce the wire has been cut and list the
1180	! set to 1 by EVENT-driven routines to tell the	1740	number of wires
1190	! the game is over and that a new one should	1750	! remaining.
1200	! returning to the top of the loop).	1760	!
1210	!	1770	CASE Live
1220	ENABLE	1780	Livewires=Livewires-1
1230	Playgame=0	1790	IF Livewires=0 THEN
1240	REPEAT	1800	CONTROL
1250	UNTIL (Playgame=1)	1810	@Sys;SET ("*NAME":
1260	!	1820	"Main/Clock")
1270	END LOOP	1830	CONTROL
1280	STOP	1840	@Sys;SET ("TIMER STATE":
1290	!	1850	"STOPPED")
1300	! This routine checks the status of the "wire"	1860	S\$="Play another
1310	! It relies on the "Wires" array to track the	1870	game?"
1320	! the wire set at any time.	1880	DIALOG
1330	!	1890	"QUESTION",S\$,Btn;SET ("TITLE":
1340	Cutwire:!	1900	"Disarmed !! ")
1350	!	1910	SELECT Btn
1360	! Check status of all ten wires.	1920	CASE 0
1370	!	1930	Playgame=1!
1380	FOR N=1 TO 10	1940	RETURN
1390	!	1950	CASE 1
1400	! Ignore the wire if it has been cut.	1960	GOTO Finis!
1410	!	1970	END SELECT
1420	IF Wires(N)<>Cut THEN	1980	ELSE
1430	!		
1440	! Otherwise, query the togglebutton value.		
1450	!		
1460	S\$=VAL\$(N)		
1470	CONTROL @Sys;SET		
1480	STATUS @Sys;RETURN		
	("VALUE":V)		

# Screen Builder

<pre>1990             ELSE 2000             CALL Pr("DANGER -- trigger wire, one left!") 2010             END IF 2020             END SELECT 2030 ! 2040 ! If you have not won or been killed, mark this wire as 2050 ! being "cut". 2060 ! 2070             Wires(N)=Cut 2080 ! 2090             END IF 2100             END IF 2110             NEXT N 2120             RETURN 2130 ! 2140 ! This routine tells you that the bomb exploded and you are dead 2150 ! 2160 Boomboom! 2170             CONTROL @Sys;SET (**NAME":"Main/Clock","TIMER STATE":"STOPPED") 2180             S\$="YOU'RE DEAD!" 2190             S\$=S\$&amp;Eol\$&amp;Eol\$ 2200             S\$=S\$&amp;"Play another game?" 2210             DIALOG "QUESTION",S\$,Btn;SET (**TITLE":" Bomb Exploded !! ") 2220             IF Btn=0 THEN 2230                 Playgame=1 2240                 RETURN 2250             ELSE 2260                 GOTO Finis 2270             END IF 2280             RETURN 2290 !</pre>	<pre>2300 ! Go here when done. 2310 ! 2320 Finis! 2330             ASSIGN @Sys TO *! Delete SYSTEM widget 2340             END 2350 ! 2360 ! ***** End of Main Program ***** 2370 ! 2380 ! Routine to print string in PRINTER widget. 2390 ! 2400             SUB Pr(S\$) 2410                 COM @Sys 2420                 CONTROL @Sys;SET (**NAME":"Main/Printer","APPEND TEXT":S\$) 2430             SUBEND</pre>
--	--