



The Excel Button Calc Workbook

By Dell Krauchi

Disclaimer

All information in this document is Copyright © 2024 Dell Krauchi and Autolt Consulting Ltd. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage or retrieval system, or otherwise, without prior permission and written consent of Dell Krauchi, Sentinel Music Studios and RML Labs. All trademarks and brand identities as used in this text are used for descriptive purposes only and remain the property of their respective owners.

Preamble

The Excel Button Calc Workbook is something I devised to assist me in developing GUICreate and GUICtrlCreateButton parameters. This text briefly explains this workbook.

Function Reference

This text will focus entirely on the syntax that is employed for two Autolt GUI Functions, namely GUICreate, and GUICtrlCreateButton. In particular, as the syntax specifically pertains to the development of a "buttons".

The two Function references consist of the following syntax:

- GUICreate ("title" [, width [, height [, left = -1 [, top = -1 [, style = -1 [, exStyle = -1 [, parent = 0]]]]]])
- GUICtrlCreateButton ("text", left, top [, width [, height [, style = -1 [, exStyle = -1]]]])

Excel

As will be discovered, Microsoft Excel will be employed throughout this present text. Though no in-depth explanation will be provided as *why* Excel was employed, suffice it to say, that for such a purpose as this, Excel *is most definitely, the preferred choice*.

Buttons	QTY			
Across:	0			
Down:	0			
GUICreate				
Width:	0			
Height:	0			
New Incremental Value	R1	R2	R3	R4
Width:	10	15	20	25
	[NincVal]			
	B1	B2	B3	B4
Height:	10	15	20	25
	[NincVal]			
GUICtrlCreateButton				
Left:	10			
Top:	10			
Width:	0			
Height:	0			
Spacer:	5			
Right Spacer:	10			
Bottom Spacer:	10			

Colour Scheme

Colours have been implemented throughout, so as to assist in the observance of the end results of the entering of the various parameters.

This colour scheme will coincide with the syntax that will be employed for the two aforementioned AutoIt GUI Functions.

Keyword	Colour
GUI Width	[Light Orange]
GUI Height	[Lime]
GUICtrlCreateButton Left	[Tan]
GUICtrlCreateButton Top	[Light Yellow]
GUICtrlCreateButton Width	[Light Green]
GUICtrlCreateButton Height	[Light Turquoise]
Spacer	[Pale Blue]
Right Spacer	[Lavender]
Bottom Spacer	[Gray-25%]

Important Notes

1. The Spacer, Right Spacer and Bottom Spacer values are not *specifically* defined in the two aforementioned AutoIt GUI Functions.
 - These three values are being provided solely for "visual" purposes – all of which will affect the "spacing" around the deployed buttons.
2. The "Left" and "Right Spacer" are "linked".
 - Entering a "Left" value, updates the values in the other two cells.
 - This "linked state" was purposefully done, as these three values *should always contain the very same values*.
3. The recommended default value of "Left" is "10".
4. The "Top" and "Bottom Spacer" are also "linked".
5. The initial value for GUICreate [width] and GUICreate [height] should be initially be "0".
 - This initial value will be updated later to coincide with the total number of columns desired.

The Example GUI Script

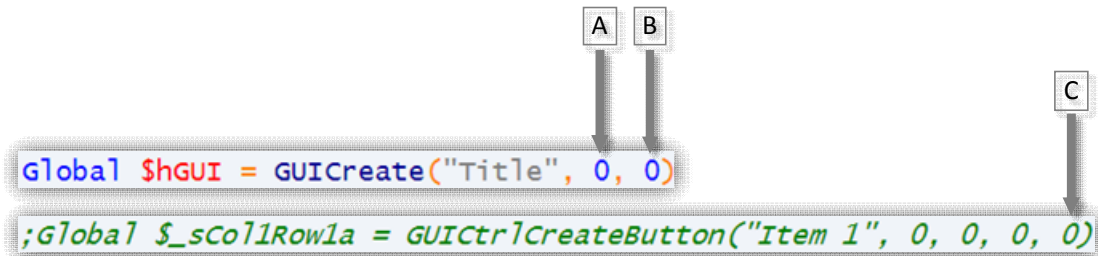
Included with this text, is a script containing a GUI *consisting of a total of four columns and four rows*, as shown below:

```

1  ;-----
2  #include <FontConstants.au3>
3  #include <GUIConstantsEx.au3>
4  ;-----
5  Global $hGUI = GUICreate("Title", 535, 135)
6  GUISetFont(14, $FW_BOLD, $GUI_FONTNORMAL, "calibri")
7  ;-----
8  ; Column 1
9  Global $sCol1Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
10 Global $sCol1Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
11 Global $sCol1Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
12 Global $sCol1Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
13 ;-----
14 ; Column 2
15 Global $sCol2Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
16 Global $sCol2Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
17 Global $sCol2Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
18 Global $sCol2Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
19 ;-----
20 ; Column 3
21 Global $sCol3Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
22 Global $sCol3Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
23 Global $sCol3Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
24 Global $sCol3Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
25 ;-----
26 ; Column 4
27 Global $sCol4Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
28 Global $sCol4Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
29 Global $sCol4Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
30 Global $sCol4Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
31 ;-----
32 GUISetState(@SW_SHOW, $hGUI)
33 ;-----
34 while 1
35     Switch GUIGetMsg()
36     Case $GUI_EVENT_CLOSE
37         ExitLoop
38     EndSwitch
39 wEnd
40 ;-----

```

The two most significant *keywords* in the above are the 1) GUICreate and 2) GUICtrlCreateButton keywords.



Important notes....

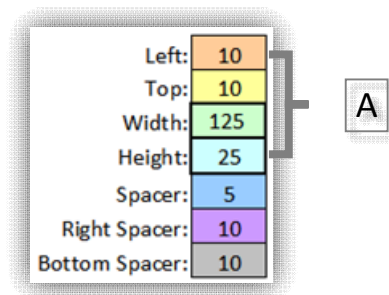
With regards to the above, please note the following:

- [A]: Refers to the GUICreate [, width] parameter [...which is Light Orange]
- [B]: Refers to the GUICreate [, height] parameter [...which is Lime]
- [C]: Refers to the GUICtrlCreateButton [, width] parameter [...which is Light Green]

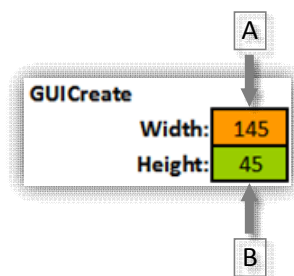
These three parameters are noted initially, as these three parameters will be employed in the following example.

Procedure [Example]

1. In the following example, a GUICreate consisting of *one column* and *one button* will be considered.
2. Enter the required GUICtrlCreateButton [left], [top], [width], [height] parameters
 - For this example, the [width] of "125" and a [height] of "25" has been entered:



3. Enter the required [Spacer], [Right Spacer], and [Bottom Spacer] parameters
 - For this example, leave the existing values "as is".
4. Make note of the following values for Column 1:
 - Row, the [,width] parameter [A].
 - Column, the [,height] parameter [B].

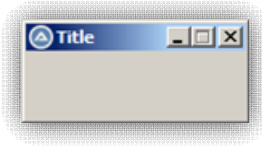


5. The following values are then entered:

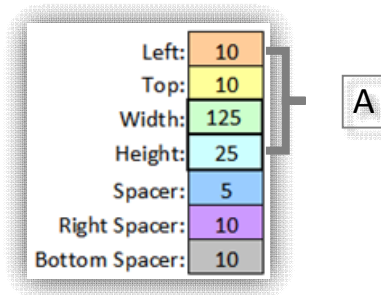
A
B

```
global $hGUI = GUICreate("Title", 145, 45)
```

6. Executing the script produces the following output:



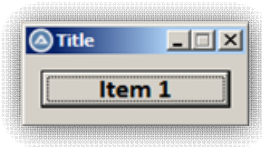
7. With the following values entered...



8. ...with the subsequent removal the "comment character"....

```
; Column 1
Global $_sCol1Row1a = GUICtrlCreateButton("Item 1", 10, 10, 125, 25)
```

9. ...the following end result is produced...



Summary

In summary thus far, a "One Column, One Row Button Example" has been thusly developed. As will be discovered, any combination of columns and rows can be developed employing the Excel Button Calc Workbook.

Limitations

The current limitations on the Excel Button Calc Workbook are 4 Columns with 4 Rows. This limitation was done purposefully, as Excel does have the applications own limits.

The next step in the "evolution" of the Excel Button Calc Workbook, is to develop a dedicated AutoIt script. If anyone happens to be at all interested, I can discuss the "math" for this "project".

Conventions

The following information is being provided for the sake of completeness:

Buttons		QTY	
Across:	0	A	
Down:	0	B	

GUICreate	
Width:	0 C
Height:	0 D

New Incremental Value				
	R1	R2	R3	R4
Width:	10	15	20	25
	[NincVal]			
	B1	B2	B3	B4
Height:	10	15	20	25
	[NincVal]			

GUICtrlCreateButton	
Left:	10 G
Top:	10 H
Width:	0 I
Height:	0 J
Spacer:	5 K
Right Spacer:	10 L
Bottom Spacer:	10 M

C D

global \$hGUI = GUICreate("Title", 0, 0)

```

; Column 1
;Global $_sCol1Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
;Global $_sCol1Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
;Global $_sCol1Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
;Global $_sCol1Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
;-----
; Column 2
;Global $_sCol1Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
;Global $_sCol1Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
;Global $_sCol1Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
;Global $_sCol1Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
;-----
; Column 3
;Global $_sCol1Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
;Global $_sCol1Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
;Global $_sCol1Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
;Global $_sCol1Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
;-----
; Column 4
;Global $_sCol1Row1a = GUICtrlCreateButton("Item 1", 0, 0, 0, 0)
;Global $_sCol1Row1b = GUICtrlCreateButton("Item 2", 0, 0, 0, 0)
;Global $_sCol1Row1c = GUICtrlCreateButton("Item 3", 0, 0, 0, 0)
;Global $_sCol1Row1d = GUICtrlCreateButton("Item 4", 0, 0, 0, 0)
;-----

```

Entry	Description	Notes	Image
[A]	Across	The number of buttons for the width of the GUI	1
[B]	Down	The number of buttons for the height of the GUI	1
[C]	Width	The width of the GUI, GUICreate ([, width]	2
[D]	Height	The height of the GUI, GUICreate ([, height]	2
[E]	Width	The value to be entered in [E] ¹	3
[F]	Height	The value to be entered in [F] ²	3
[G]	Left	The [Left] spacing	4
[H]	Top	The [Top] spacing	4
[I]	Width	The [Width] of the button	4
[J]	Height	The [Height] of the button	4
[K]	Spacer	The [Spacer] between the button ³	4
[L]	Right Spacer	The [Right] spacing	4
[M]	Bottom Spacer	The [Bottom] spacing	4

¹Positions the button positioned to the right of the existing left button

²Positions the button positioned below the existing upper button

³Either vertically or horizontally

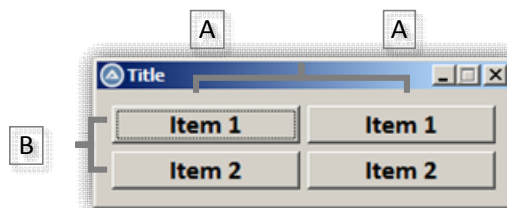


Image 1

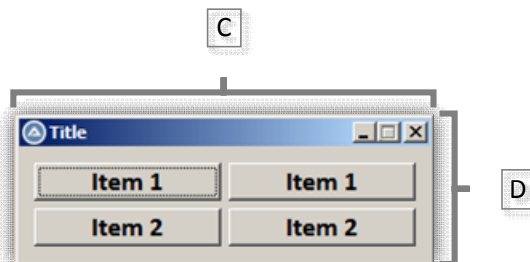


Image 2

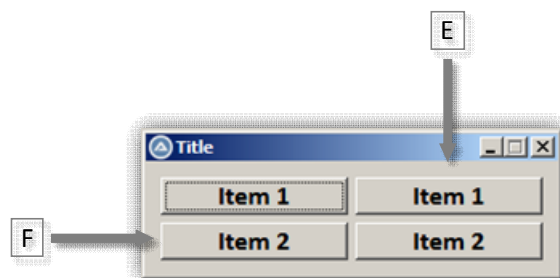
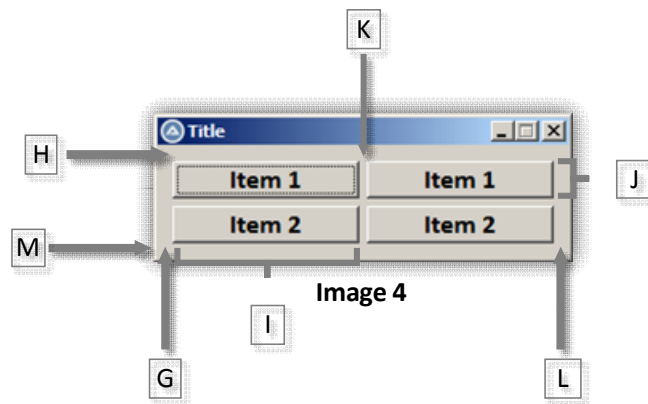


Image 3



In closing...

As always, I sincerely hope that the above information will be of some practicable use to others?

Sincerely,

Dee L

Date: November 28, 2024

Update: