

Form Generator

Contents table

1	Form generator.....	2
1.1	Calling formHandler.....	2
1.2	Data description.....	2
1.2.1	Field Name.....	2
1.2.2	Field Label.....	2
1.2.3	Length.....	2
1.2.4	Type	2
1.2.5	Default value.....	3
1.2.6	Extra.....	3
1.3	Returned Values.....	3
1.4	Controls	3
1.5	CallBack.....	3
1.5.1	CallBack function.....	3
1.5.2	CallBack button.....	4
1.6	Example.....	4
1.7	Remarks.....	4
1.7.1	Data presentation.....	4
1.7.2	Handling Buttons.....	4
2	Others functions and utilities.....	5
2.1	Retrieve Data.....	5
2.2	Retrieve Data item.....	5
2.3	Mask comma and semicolon.....	5
2.4	Display dictionary.....	5
2.5	The Sand Box.....	5
3	History.....	5
4	Technical notes.....	5
4.1	Global variables.....	5
4.2	Dictionary.....	5
4.2.1	Buttons.....	6
4.2.2	Values- key.....	6
5	My be in future.....	6
6	Alphabetical Index.....	7

Disclaimer

This SOFTWARE PRODUCT is provided by El Condor "as is" and "with all faults." El Condor makes no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies, typographical errors, or other harmful components of this SOFTWARE PRODUCT. There are inherent dangers in the use of any software, and you are solely responsible for determining whether this SOFTWARE PRODUCT is compatible with your equipment and other software installed on your equipment. You are also solely responsible for the protection of your equipment and backup of your data, and El Condor will not be liable for any damages you may suffer in connection with using, modifying, or distributing this SOFTWARE PRODUCT.

You can use this SOFTWARE PRODUCT freely, if you would you can credits me in program comment:

El Condor – CONDOR INFORMATIQUE – Turin

Comments , suggestions and critisms are welcomed: mail to rossati@libero.it

1 Form generator

The script `formHandler` is a AutoIt script (version v3.3.8.0), which allows to build and handle forms; it is sufficiently generalised for a wide use.

1.1 Calling formHandler

The function `formHandler` has three parameters, the first is the title of the generated form, the second is a list of field's description separated by semicolon and the third, optional, is a handle to an existent window or a name of a `CallBack` function (see `Remarks` section).

Without the third parameter `formHandler` is called as `PopUP`, otherwise the form is created in the parent window which must be `ACTIVE` (via `GUISetState()`).

1.2 Data description

Every field is characterised by a list of attributes which are comma separated and in this order: Field Name, Field Label, Length, Type, Default Value and Extra.

1.2.1 Field Name

Is the name of the field, which is returned, when the form is closed, with the value associated.

1.2.2 Field Label

Label of Field or caption of button, if omitted is used Field Name.

1.2.3 Length

The length, in characters, of the field or of the buttons for Type **BL**. For Type **U** is ignored and substituted by the length of the `default` attribute.

1.2.4 Type

- Buttons:
 - **B** button;
 - **BL** button List, every button will have a Label Field as caption;
 - **BC** CallBack button;
 - **R** radio button, a set of Radio buttons.
- **CKB** check box, values are 1 for selected check boxes, 4 otherwise; if the default value is 1 then the check box appears checked.
- Combos:
 - **CMB** combo box.
 - **CMT** combo box modifiable, e.g. the user can input text or select from combo.
- Text fields:
 - **F** file;
 - **D** folder;
 - **N** numeric field (right aligned);
 - **P** password field;
 - **T** text field is the default if the Type is omitted.
 - **U** not modifiable field.

An **M** after Type means that the field is mandatory, this is ignored for type **Bx**, and **U**.

The Types are accepted also in lower case.

1.2.5 Default value

This value can be used for Text Fields, Combos, Check boxes, Buttons and CallBack Buttons; it is restored when the `Reset` button is clicked. It can also be used to display data (Type `U`).

For buttons type `B` and `BC` is the value returned if the button is clicked.

1.2.6 Extra

For `combos` and `Button List` is a item list separated by `|`, a couple of `| (| |)` insert an empty item.

For Type `BC` is a name of a function which is called when the button is clicked.

For Type `F` can be a filter file e.g. `Images (*.jpg;*.bmp)|Videos (*.avi;*.mpg)`

For `combos` or `button List` if you would return a code associated to the description, the item has the form: `key,value`.

Examples:

```
Unit,Measure Unit,5,CMTM,Kg,||Kg,Kilos|Lt,Liters|Mc,CubicMeters|Wh,Watt/hour
Field0,Field 0,15,FM,,Images (*.jpg;*.bmp)|Videos (*.avi;*.mpg)1
```

1.3 Returned Values

The function return a two dimension array where the column 0 (0 index) contains the field Names and the column 1 contains the values; if the form is cancelled an empty string is returned.

The value returned for buttons Type `B` or `BC` is the value indicated in the `default` field; for `Button List (BL)` is the value associated, tacked from `extra` field.

1.4 Controls

Numeric fields, if the type is `N` (comma and sign aren't accepted).

1.5 CallBack

1.5.1 CallBack function

`formHandler` works in polling mode that is with `GUIOnEventMode=0`; this means which the events of the parent window aren't handled; if you need to handle also these events, you must use a `CallBack` function.

For handle Events of the controls external to `formHandler`, they must be coded in a function, besides the identifiers of controls must be declared `Global` and the third parameter of `formHandler` is the name of that function which is called if the event is not an event of `formHandler`; the script below is a sample:

```
#Include <Array.au3>
#include <WindowsConstants.au3>
#include "formhandler.au3"
$parentWindow = GUICreate("Parent Window", 450, 350, 100, 100)
$helpMenu = GUICtrlCreateMenu("?")
Global $infoItem = GUICtrlCreateMenuItem("&Info", $helpMenu)
Global $seeItems = GUICtrlCreateMenuItem("&See Data", $helpMenu)
AutoItSetOption("GUISizeMode", $GUI_DOCKALL)
$records = "Name,,12,CMB,Beta,||K1,Alfa|K2,Beta|K3,Delta" _
           & " ;Nome2,Ok,7,BL,,K1,Alfa|K2,Beta|K3,Delta;Nome3,ClickMe,7,B,,Gamma"
$resp = formHandler("Form handler sample",$records,"checkEvents")
_ArrayDisplay($resp,"Data")
Do
    $msg = GUIGetMsg()
    checkEvents($msg)
```

¹ This field must be masked, e.g. by mask function so the semicolon is translated to 0x0032:

```
"Field0,Field 0,15,FM,," & mask("Images (*.jpg;*.bmp)|Videos (*.avi;*.mpg)")
```

```

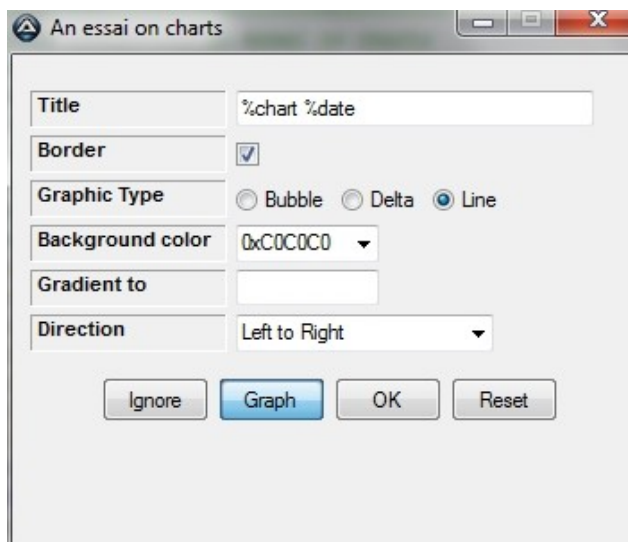
Until $msg = $GUI_EVENT_CLOSE
Func checkEvents($msg)
    If $msg = $infoItem Then
        MsgBox(0, "Credits", "Condor Informatique - Turin" _
            & @CRLF & "Form Handler Sample" _
            & @CRLF & "Program developped with AutoIt3")
    EndIf
    If $msg = $seeItems Then
        $resp = GetData()
        _ArrayDisplay($resp, "Data")
    EndIf
EndFunc

```

1.5.2 CallBack button

A CallBack Button (Type **BC**) is useful for retrieve data without closing the form. When clicked, formHandler call the function indicated in the `extra`, field passing the array of data, without destroying a form.

1.6 Example



```

Title,,25,T,%chart %date;
Border,,10,CKB,1;
Type,Graphic Type,10,R,Line,Bubble|
Delta|Line;
Graph,,10,BC,Graph,makeGraph;
BColor,Background
color,10,CMT,0xC0C0C0,0xC0C0C0|
0xc080c0|0xa0a0a0;
BColorG,Gradient to,10,T;
Direction,,18,CMT,Left to Right,0,Left
to Right|45,Bottom Left to Top Right|
90,Bottom to Top|135,Top Left to Bottom
Right

```

1.7 Remarks

1.7.1 Data presentation

The data are presented in the order they appear in the parameters list, with the exception of the Type **B** and **BC** buttons that appear together to buttons inserted by formHandler at the bottom of the form.

For widget of Type Text (**T**, **F**, **D**, **U**, **P**) if the length exceed 50 the widget is multi lined.

1.7.2 Handling Buttons

formHandler inserts the `Ok` button, the `Ignore` button and the `Reset` button, this is function of the widgets contained in the form:

- the `Ignore` button is always present,
- the `Reset` button is present if there are data fields (e.g. Type **F**, **D**, **N**, **P**, **R**, **CKB**, **CMB**, **CMT**) ,
- the `Ok` button is present if there are only data fields, in other words Buttons and Buttons List replace the button `Ok`.

2 Others functions and utilities

2.1 Retrieve Data

The function `getData()` is used internally for return data; it can be called also for retrieve data without closing the form (see example above).

2.2 Retrieve Data item

The function `TakeData($array,$key)`² can be used for retrieve data from the returned array by name instead of by position:

Example:

```
$records = "BackupName,Backup Name,15,TM;FileName,File Name,40,FM;...  
$aReturn = formHandler("Backup",$records)  
MsgBox(0,"",TakeData($aReturn,"FileName"))
```

2.3 Mask comma and semicolon

If `label` or `default` or `extra` contains commas or semicolons, the function `mask($data)` return a string with that characters encoded.

2.4 Display dictionary

The function `displayDict($dict)`, contained in `functions.au3`, display a dictionary content.

2.5 The Sand Box

The script `sbFormHandler.au3` help to understand, build parameters and test `formHandler`.

3 History

0.2.7 text multi line when length is greater 50,
comma and semicolon accepted in label, default and item list,
comments added.

0.3.0 Added Radio Button
added CallBack Button

4 Technical notes

4.1 Global variables

The global variables of `formHandler` are:

- `$afhRecords` array of data fields
- `$fhDict` dictionary

4.2 Dictionary

The dictionary is a COM object used create a key value access for handle internal activities:

```
Global $fhDict = ObjCreate("Scripting.Dictionary")
```

²This function works also on the array returned by `IniReadSection` function.

4.2.1 Buttons

Type	Key	Value
Button	Control ID	Type BB, Index to data description array, value
Button List	Control ID	Type BL, Index to data description array, value
OK Button	Control ID	Type BO
Reset Button	Control ID	Type BR
Ignore Button	Control ID	Type BI
CallBack Button	Control ID	Type BCB, Index to data description array, default value, name of callback function
Radio button	Control ID	Type R, Radio label, receiver ControlID
Radio button	NameField	ControlID of field to store value

4.2.2 Values- key

For Combos, Radiobuttons and Button list there is an entry in Dictionary for each item in extra field:

Field Name & Item Value	Key value
-------------------------	-----------

5 My be in future

- Add Hidden Fields
- handle CR and LF for Type U,
- field controls (e.g. range numeric, date, mail, etc.).

6 Alphabetical Index

Button.....	
Button List.....	2
Button position.....	4
Caption.....	2
Ignore button.....	4
OK button.....	4
Reset button.....	3, 4
CallBack.....	2, 3
Default value.....	3
Field.....	
Control.....	
Numeric.....	3
Label.....	2
Length.....	2
Length exceed.....	4
Mandatory.....	2
Name.....	2
File.....	
Filter.....	3
Function.....	
CallBack.....	3
Mask comma and semicolon.....	5
Retrieve Data.....	5
Retrieve data item.....	5
Type.....	
Button.....	2
Check Box.....	2
Combo Box.....	2
File.....	2
Folder.....	2
Not modifiable.....	2
Radio button.....	2