

Form & Control Sizer - Version 1.0

Content

Generalities

Tips & Tricks

Properties

This VBX control is design to help the programmer in the design of the screen's layout. Many functionalities ar available:

- Save an load form's size, position and satus in an user initialization file. SDI forms, MDI parent and child forms are handled.
- Move and resize a form in the screen or in a MDI parent window. May consider the iconized windows area at the bottom of the screen or of MDI parent window.
- Move and resize a control in a form or in another control.
- Consider automatically all elements like screen's resolution, window's or control's size, border with, caption bar, menu, ...
- May contain a child control witch can be resized automatically. So, you can handle the size and the position of any other control of any type.

In addition:

- To give an usefull interface to the intitialization file's access.

All these operations are performed without any line of code. Nevertheless, you can disable the automatic handling and customize easily your layout.



Form & Control Sizer - Version 1.0

Generalities

Fell free to contact me for comments and suggestions

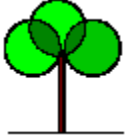
Copyright

License

Registration

System requierments

Technical support



Form & Control Sizer - Version 1.0

Copyright

This "**Form & Control Sizer**" is Copyrighted by **Michel Lombart**. All rights are reserved.

All marks, names and softwares mentioned in this software are registered or copyrighted by the owners or the authors.

Please, read carefully the [license](#).



Form & Control Sizer - Version 1.0

License

YOU SHOULD READ CAREFULLY THE FOLLOWING TERMS AND CONDITIONS BEFORE USING OR DISTRIBUTING THIS SOFTWARE. USING OR DISTRIBUTING THIS SOFTWARE INDICATES ACCEPTANCE OF THESE TERMS AND CONDITIONS.

DEFINITIONS OF TERMS

In this document, the software files, the documentation and all related files distributed with the software are called "the package".

The term "the author" refer to the author of this software, **Michel Lombart**.

COPYRIGHT

This package is copyrighted and remain the property of the author. All rights are reserved.

You may not sublicense, lease, rent, convey, translate, modify, decompile, convert to another programming language, reverse engineer or disassemble the package or any file for any purpose.

DISTRIBUTION

This package is distributed originally via electronic file transfer through Internet. All files are archived and compressed in one file for ease of distribution. This file is the only official and was compiled in june 1996.

You may freely distribute this package to others via an electronic transfer mechanism or electronic storage media including, but not limited to, diskette, hard disk, removable disk, tape or optical devices such as CD-ROM or WORM provided that no files are deleted from the package or are modified in any way. No files may be added to the package. You may place the package into any archive or compressed format for ease of distribution.

If this package is distributed on a commercial support, you must declare, before the sale, that this package is distributed under the shareware concept and that the purchaser must pay the registration fee to the author.

You may not distribute the package to others under any circumstances.

LICENSE

After your registration and payment, a personal registration key will be sent to you by E-Mail. You will have also free minor upgrades, reduced fee for major upgrades registration and a free access to the [technical support](#). Nevertheless, there is no guarantee that upgrade, neither minor nor major, will be provided in the future.

You may use the package on only one computer simultaneously. If you need to use the package on more than one computer simultaneously, please contact the author for information about site licenses.

You are not obtaining title to the package or any copyright rights. You may not sublicense, lease, rent, convey, translate, modify, decompile, convert to another programming language, reverse engineer or disassemble the package or any file for any purpose.

A licensed user is free to distribute the SRTSIZER.VBX file, and only this one, with its own softwares. It's absolutely forbidden to distribute the registration key in any form, including the license file. The shareware author using this VBX in his own software must enclose, in his copyright notice, the following clause: "The srtsizer.vbx is copyrighted by Michel Lombart".

LIMITED WARRANTY

THE SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE AUTHOR DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ALL SUCH WARRANTIES ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED.

YOUR EXCLUSIVE REMEDY AND AUTHOR'S ENTIRE LIABILITY ARISING FROM OR IN CONNECTIONS WITH THE PACKAGE AND/OR THIS LICENSE (INCLUDING WITHOUT LIMITATION FOR BREACH OF WARRANTY OR INFRINGEMENT) SHALL BE, AT AUTHOR'S OPTION, THE REFUND OF REGISTRATION FEES.

IN NO EVENT THE AUTHOR OF THIS SOFTWARE SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE EVEN IF AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS.



Form & Control Sizer - Version 1.0

Registration

PLEASE, READ CAREFULLY THE LICENSE BEFORE THE REGISTRATION.

Registration fee

The registration fee for this product is 20 US\$.

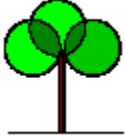
Ordering instructions

After sending your registration form and fee, you will receive by E-Mail a registration key. This key will be introduced in the dialog box activated by the registration software included in this package. You will receive any instructions by E-Mail. The typical delay before receiving your key is about one week, maybe two during hollidays.

Please, ***take care to send your exact first name and last name*** (no case sensitive). These information are coded in the registration key.

Since this version, all Saint Roch Tree softwares must be licensed to the same user on each computer.

Click [here](#) to achieve the registration procedure.

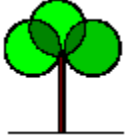


Form & Control Sizer - Version 1.0

System requirements

This control is designed to run under Visual Basic Version 2 at least.

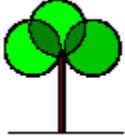
Of course, like all VBX controls, you must distribute the SRTSIZER.VBX with your application.



Form & Control Sizer - Version 1.0

Technical support

Registered users can access to a free technical support. This support is only available by E-Mail. Simply, complete the form you will find in my [web page](#).



Form & Control Sizer - Version 1.0

Tips & Tricks

Most of the capabilities are explained in [the demonstration software](#). Here, there are some additional explanations.

Management of forms and controls:

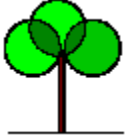
[Definitions of container and child items](#)

Access to the initialization files:

[General specifications](#)

[Automatic access, save and load position and status of a form](#)

[Customized access, save and load integer and string variables](#)



Form & Control Sizer - Version 1.0

Handle of container and child items

Some definitions are used: (refer to related properties for more informations)

The Sizer Control:

It's the Saint Roch Tree Form & Control Sizer itself. You can load any copy of it in your forms.

The Bounded Item:

It's the object bounded by the sizer control. It can be the sizer control itself, his parent control, his parent window or the top parent window. This last option is used when the sizer control is loaded in a MDI child Form and when you want bound the MDI parent window. You define the Bounded Item by setting the BoundedItem property to the appropriate value.

The bounded item can be automatically sized or moved by setting the AutoAction1, AutoAction2 and AutoActionValid properties to the appropriate value. You can also force at run time the size or the move of the bounded control by setting the Action to the desired value.

The Container Item:

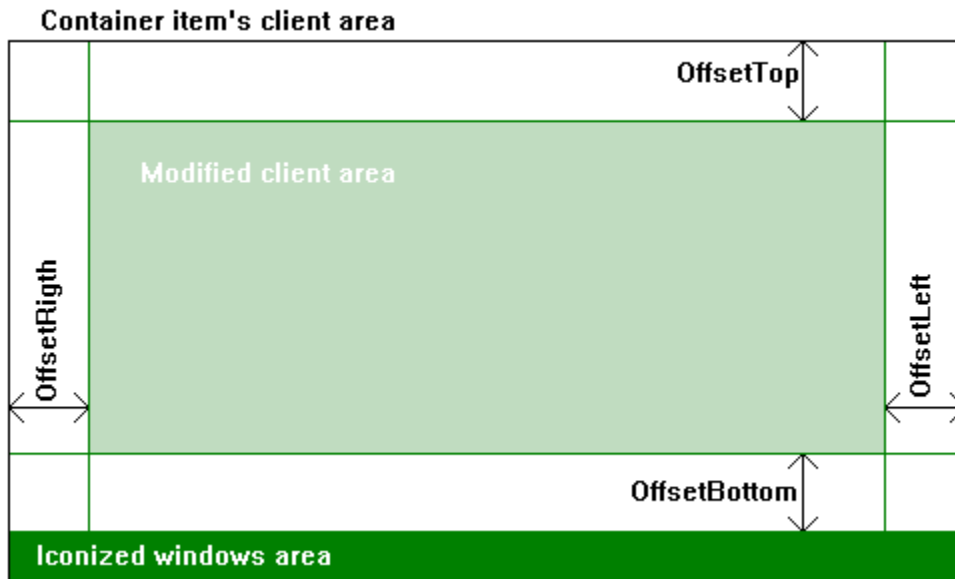
It's the bounded item's parent object. It would be the screen, a form or another control like a picture control for instance.

All moving and sizing operations affect the bounded item relating to the container item.

The child Item:

It's an item contained in the sizer control. This item can be resized automatically to fit the sizer control size. So, you can handle the size and the position of another control. You just need to load a sizer control, to insert in it the control needed and to declare that the bounded item is the sizer control itself. This option is active when the AutoResizeChild property is set to TRUE.

When moving and sizing operations are performed, the sizer control use the container item's client area. This area is modified accordingly setup properties:



- **ConsiderIconArea:**

By setting that property to TRUE, the client area is limited to the top of the icon area at the bottom of the container item. This property is valid only if the container item is the screen or a MDI parent window.

- **OffsetLeft, OffsetRight, OffsetTop and OffsetBottom:**

Those properties modify the container item client area. They are valid according the **Action** performed. Look to the **Action** property specifications for more informations. Thoses values may be negative.

The modified container item's client area, like the original client area, is only a reference system. There are no limits to the position and to the size of the bounded item. For instance, it would be greater than the modified client area. The position and the size depend of the type of the **Action** and how that **Action** considers the offset values.



Form & Control Sizer - Version 1.0

Access to the initialization files - General specifications

Structure of an initialization file

Accessing to an initialization file is very simple, as well for an automatic form's position handling as for setting variables values.

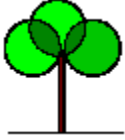
You have a minimal setting to access to the initialization file:

First, you must decide where the initialization file will be located. You set the IniFileDirectory to your choice. You can choice to locate the initialization file as well in the windows directory as in the application directory.

Second, you must define the name of the initialization file. Simply set the IniFileName to the name of the file.

Now, there are two possibilities:

Automatic access, save and load position and status of a form
Customized access, save and load integer and string variables



Form & Control Sizer - Version 1.0

Access to the initialization files - Automatic handling

With the Saint Roch Tree Form and Control Sizer, you can handle form's position, size and status without any line of code. However with some commands, you can easily customize your application.

First, you determine the section's name where the informations will be loaded. You must define one section name by form used in your project. It's a good and simple practice to set the [IniAutoSectionName](#) property to the name of the form.

If you want an absolutely automatic handling, set the [AutoLoadPosition](#) and [AutoSavePosition](#) properties to TRUE. That all! Now, the form's position, size and status will be automatically saved and loaded each time you will load or unload the form.

Here, there are some examples of customization:

If you want to force the form to be loaded every time with a normal status whatever with the status at the precedent unload, simply, set the [AutoForcedState](#) property to the value **2 - NORMAL**.

If you want to restore a predefined setting whatever all the modifications, set the [AutoSavePosition](#) property to FALSE. You can insert a "Save position" item in your menu. That option will force a **SZ_ACTION_SAVEPOSITION** [Action](#) so the user can save the form's position, size and status only if he want.



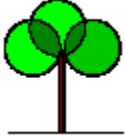
Form & Control Sizer - Version 1.0

Access to the initialization files - Customized handling

To access to the initialization file defined, first, set the [IniVariableSectionName](#) and [IniVariableKeyName](#) properties to the correct strings.

To write a string or an integer to the initialization file, set the [IniVariableStringValue](#) property or the [IniVariableIntegerValue](#) property to the correct value. Then, force the action **SZ_ACTION_SAVESTRING = 201** or **SZ_ACTION_SAVEINTEGER = 203**.

To read a string or an integer from the initialization file, force the action **SZ_ACTION_LOADSTRING = 202** or **SZ_ACTION_LOADINTEGER = 204**. The readed value is in the [IniVariableStringValue](#) property or the [IniVariableIntegerValue](#) property.



Form & Control Sizer - Version 1.0

Properties

Only specific properties are documented. They can be read and written at both design and run time excepted the Action property which is accessible only at run time and is write only.

Others properties are standard Visual Basic properties.

Action
AutoAction1
AutoAction2
AutoActionValid
AutoForcedState
AutoLoadPosition
AutoResizeChild
AutoSavePosition
BorderStyle
BoundedItem
ConsiderIconArea
Height
Hwnd
Index
IniAutoSectionName
IniFileDirectory
IniFileName
IniVariableIntegerValue
IniVariableKeyName
IniVariableSectionName
IniVariableStringValue
Left
Name
OffsetBottom
OffsetLeft
OffsetRight
OffsetTop
Tag
Top
Visible
Width



Form & Control Sizer - Version 1.0

Action Property

Object:

Determines an action to be performed.

Remarks:

This property is accessible only at run time and is write only.

Setting:

Moving and sizing actions.

1 - Move horizontally the bounded item to the middle of the container item. OffsetLeft and OffsetRight values are considered.

2 - Move Vertically the bounded item to the middle of the container item. OffsetTop and OffsetBottom values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the ConsiderIconArea property is TRUE.

3 - Move horizontally and vertically the bounded item to the middle of the container item. OffsetLeft, OffsetRight, OffsetTop and OffsetBottom values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the ConsiderIconArea property is TRUE.

4 - Move the bounded item to the top of the container item. The OffsetTop value is considered.

5 - Move the bounded item to the bottom of the container item. The OffsetBottom value is considered. If the container item is the screen or a MDI parent window, the icon area is considered if the ConsiderIconArea property is TRUE.

6 - Move the bounded item to the left of the container item. The OffsetLeft value is considered.

7 - Move the bounded item to the right of the container item. The [OffsetRight](#) value is considered.

8 - Move the bounded item to the topleft corner of the container item. The [OffsetLeft](#) and [OffsetTop](#) values are considered.

9 - Move the bounded item to the the middle of the top edge of the container item. The [OffsetLeft](#), [OffsetRight](#) and [OffsetTop](#) values are considered.

10 - Move the bounded item to the topright corner of the container item. The [OffsetRight](#) and [OffsetTop](#) values are considered.

11 - Move the bounded item to the middle of the left edge of the container item. The [OffsetLeft](#), [OffsetBottom](#) and [OffsetTop](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

12 - Move the bounded item to the middle of the right edge of the container item. The [OffsetRight](#), [OffsetBottom](#) and [OffsetTop](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

13 - Move the bounded item to the bottomleft corner of the container item. The [OffsetLeft](#) and [OffsetBottom](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

14 - Move the bounded item to the the middle of the bottom edge of the container item. The [OffsetLeft](#), [OffsetRight](#) and [OffsetBottom](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

15 - Move the bounded item to the bottomright corner of the container item. The [OffsetTop](#) and [OffsetBottom](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

16 - Tile horizontally the bounded item in the container item. [OffsetLeft](#) and [OffsetRight](#) values are considered.

17 - Tile vertically the bounded item in the container item. [OffsetTop](#) and [OffsetBottom](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE.

18 - Tile vertically and horizontally the bounded item in the container item. [OffsetLeft](#), [OffsetRight](#), [OffsetTop](#) and [OffsetBottom](#) values are considered. If the container item is the screen or a MDI parent window, the icon area is considered if the [ConsiderIconArea](#) property is TRUE

Loading and saving form's position, size and status actions.

Thoses actions are valid only if the bounded item is a form. Otherwise, no action is performed. An error code is not generated.

The directory and the name of the initialization file are defined in the [IniFileDirectory](#) and [IniFileName](#) properties. The section name is defined in the [IniAutoSectionName](#) property.

101 - Save the bounded item position, size and status in the initialization file.

102 - Load the bounded item position, size and status from the initialization file. The staus of the form maybe modified accordingly the value contained in the [AutoForcedState](#) property.

Accessing initialization file action.

The directory and the name of the initialization file are defined in the [IniFileDirectory](#) and [IniFileName](#) properties. The section name and the key name are defined in the [IniVariableSectionName](#) and [IniVariableKeyName](#) properties.

201 - Save in the initialization file the string contained in the [IniVariableStringValue](#) property.

202 - Load from the initialization file a string witch will be accessible by reading the [IniVariableStringValue](#) property. If there are no valid string in the initialization file, the

IniVariableStringProperty will be empty.

203 - Save in the initialization file the integer contained in the [IniVariableIntegerValue](#) property.

202 - Load from the initialization file an integer which will be accessible by reading the [IniVariableIntegerValue](#) property. If there are no valid integer in the initialization file, the IniVariableIntegerProperty will be 0.

Auto action:

301 - Executes, when a sizer control is already loaded, the action memorized in the [AutoAction1](#) property.

302 - Executes, when a sizer control is already loaded, the action memorized in the [AutoAction2](#) property.

303 - Executes, when a sizer control is already loaded, the actions memorized in the [AutoAction1](#) and [AutoAction2](#) properties.

Constants defined in the SRTSIZER.BAS file are:

Global Const SZ_ACTION_HMIDDLE = 1
Global Const SZ_ACTION_VMIDDLE = 2
Global Const SZ_ACTION_HVMIDDLE = 3
Global Const SZ_ACTION_TOP = 4
Global Const SZ_ACTION_BOTTOM = 5
Global Const SZ_ACTION_LEFT = 6
Global Const SZ_ACTION_RIGHT = 7
Global Const SZ_ACTION_TOPLEFT = 8
Global Const SZ_ACTION_TOPMIDDLE = 9
Global Const SZ_ACTION_TOPRIGHT = 10
Global Const SZ_ACTION_MIDDLELEFT = 11
Global Const SZ_ACTION_MIDDLERIGHT = 12
Global Const SZ_ACTION_BOTTOMLEFT = 13
Global Const SZ_ACTION_BOTTOMMIDDLE = 14
Global Const SZ_ACTION_BOTTOMRIGHT = 15
Global Const SZ_ACTION_HTILE = 16
Global Const SZ_ACTION_VTILE = 17
Global Const SZ_ACTION_HVTILE = 18

Global Const SZ_ACTION_SAVEPOSITION = 101
Global Const SZ_ACTION_LOADPOSITION = 102

Global Const SZ_ACTION_SAVESTRING = 201
Global Const SZ_ACTION_LOADSTRING = 202
Global Const SZ_ACTION_SAVEINTEGER = 203
Global Const SZ_ACTION_LOADINTEGER = 204

Global Const SZ_ACTION_AUTO1 = 301
Global Const SZ_ACTION_AUTO2 = 302
Global Const SZ_ACTION_AUTO12 = 303

Data type:

Integer



Form & Control Sizer - Version 1.0

AutoAction1 & AutoAction2 Properties

Object:

Determines the Actions to be performed according the AutoActionValid property or by setting the Action to values between 301 and 303.

Remarks:

AutoAction1 is performed before AutoAction2.

Setting:

All move and size Actions. (values between 0 to 18)

Default:

0 - NONE

Data type:

Integer (Enumerated)



Form & Control Sizer - Version 1.0

AutoActionValid Property

Object:

Determines when the actions specified in the [AutoAction1](#) and [AutoAction2](#) properties must be automatically performed.

Setting:

- 0 - NEVER** - [AutoActions](#) are not automatically performed
- 1 - AT LOADING** - [AutoActions](#) are performed at the loading of the control
- 2 - ALWAYS** - [AutoActions](#) are performed every time that the container item is moved or resized.

Constants defined in the SRTSIZER.BAS file are:

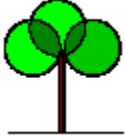
```
Global Const SZ_AUTOACTIONVALID_NEVER = 0
Global Const SZ_AUTOACTIONVALID_ATLOADING = 1
Global Const SZ_AUTOACTIONVALID_ALWAYS = 2
```

Default:

0 - NEVER

Data type:

Integer (Enumerated)



Form & Control Sizer - Version 1.0

AutoForcedState Property

Object:

Determines the visibility state of a form when the initialization file is read.

Remarks:

This property is valid only when the bounded item is a form.

Setting:

- 0 - NONE - The form is loaded with the memorized state**
- 1 - MINIMIZED - The form is loaded with a minimize state**
- 2 - NORMAL - The form is loaded with a normal state**
- 3 - MAXIMIZED - The form is loaded with a maximize state**

Constants defined in the SRTSIZER.BAS file are:

```
Global Const SZ_AUTOFORCEDSTATE_NONE = 0
Global Const SZ_AUTOFORCEDSTATE_MINIMIZED = 1
Global Const SZ_AUTOFORCEDSTATE_NORMAL = 2
Global Const SZ_AUTOFORCEDSTATE_MAXIMIZED = 3
```

Default:

0 - NONE

Data type:

Integer (Enumerated)



Form & Control Sizer - Version 1.0

AutoLoadPosition Property

Object:

Determines whether the position and the status are readed from the initialization file when a form is loaded.

Remarks:

This property is valid only when the bounded item is a form.

If there are unvalid informations in the initalization file, the position and the status at design time are used.

Setting:

FALSE - The position and the status of the loaded form are not read from the initialization file.

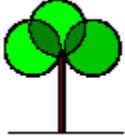
TRUE - The position and the status of the loaded form are read from the initialization file.

Default:

FALSE - The position and the status of the loaded form are not read from the initialization file.

Data type:

Integer (Boolean)



Form & Control Sizer - Version 1.0

AutoResizeChild Property

Object:

Determines whether a child control must be resized automatically.

Remarks:

Only the first child control is resized.

Setting:

**FALSE - Child control is not automatically resized.
TRUE - The first child control is automatically resized.**

Default:

FALSE - Child controls are not automatically resized.

Data type:

Integer (Boolean)



Form & Control Sizer - Version 1.0

AutoSavePosition Property

Object:

Determines whether the position and the status are memorized in the initialization file when a form is unloaded.

Remarks:

This property is valid only when the bounded item is a form.

Setting:

FALSE - The position and the status of the unloaded form are not memorized in the initialization file.

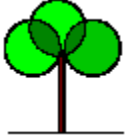
TRUE - The position and the status of the unloaded form are memorized in the initialization file.

Default:

FALSE - The position and the status of the unloaded form are not memorized in the initialization file.

Data type:

Integer (Boolean)



Form & Control Sizer - Version 1.0

BoundedItem Property

Object:

Determines the bounded item.

Remarks:

If a selection is invalid, the highest valid selection is considered.

Setting:

- 0 - Control**
- 1 - Parent Control**
- 2 - Parent Window**
- 3 - Top Parent Window**

Constants defined in the SRTSIZER.BAS file are:

```
Global Const SZ_BOUNDED_CONTROL = 0  
Global Const SZ_BOUNDED_PARENTCONTROL = 1  
Global Const SZ_BOUNDED_PARENTWINDOW = 2  
Global Const SZ_BOUNDED_TOPPARENTWINDOW = 3
```

Default:

- 0 - Control**

Data type:

Integer (Enumerated)



Form & Control Sizer - Version 1.0

ConsiderIconArea Property

Object:

Determines whether the icon area must be considered.

Remarks:

This option is valid only if the container item is the screen or a MDI parent window.

Setting:

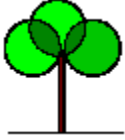
**FALSE - Icon area is not considered.
TRUE - Icon area is considered.**

Default:

FALSE - Icon area is not considered.

Data type:

Integer (Boolean)



Form & Control Sizer - Version 1.0

IniAutoSectionName Property

Object:

Determines the name of the section where informations about size, position and status of the form are memorized in the initialization file.

Setting:

The name of the section.

Default:

"Form"

Data type:

String



Form & Control Sizer - Version 1.0

IniFileDirectory Property

Object:

Determines the directory of the initialization file.

Setting:

0 - Application Directory - The initialization file is in the application directory.

1 - Windows Directory - The initialization file is in the windows directory.

Constants defined in the SRTSIZER.BAS file are:

Global Const SZ_INIFILEDIRECTORY_APPLICATION = 0

Global Const SZ_INIFILEDIRECTORY_WINDOWS = 1

Default:

0 - Application Directory

Data type:

Integer (Enumerated)



Form & Control Sizer - Version 1.0

IniFileName Property

Object:

Determines the name of the initialization file.

Remarks:

Only a valid file name is allowed. The extension is forced to .ini.

Setting:

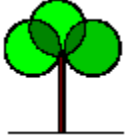
The name of the initialization file.

Default:

"project.ini"

Data type:

String



Form & Control Sizer - Version 1.0

IniVariableIntegerValue Property

Object:

Determines a variable integer value to be written or readed from the intialization file.

Setting:

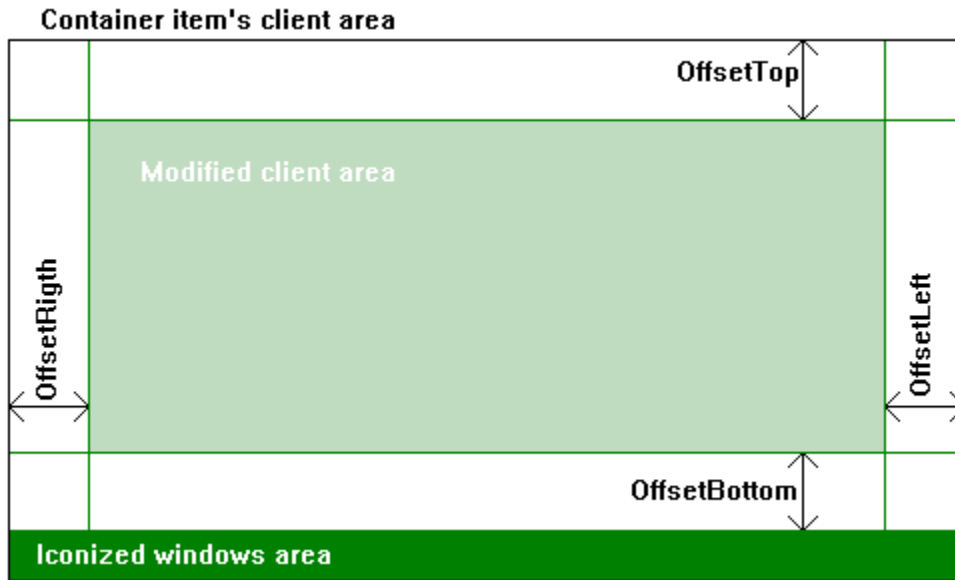
The integer value.

Default:

0

Data type:

Integer



Form & Control Sizer - Version 1.0

IniVariableKeyName Property

Object:

Determines the key name of a variable to be written or readed from the intialization file.

Setting:

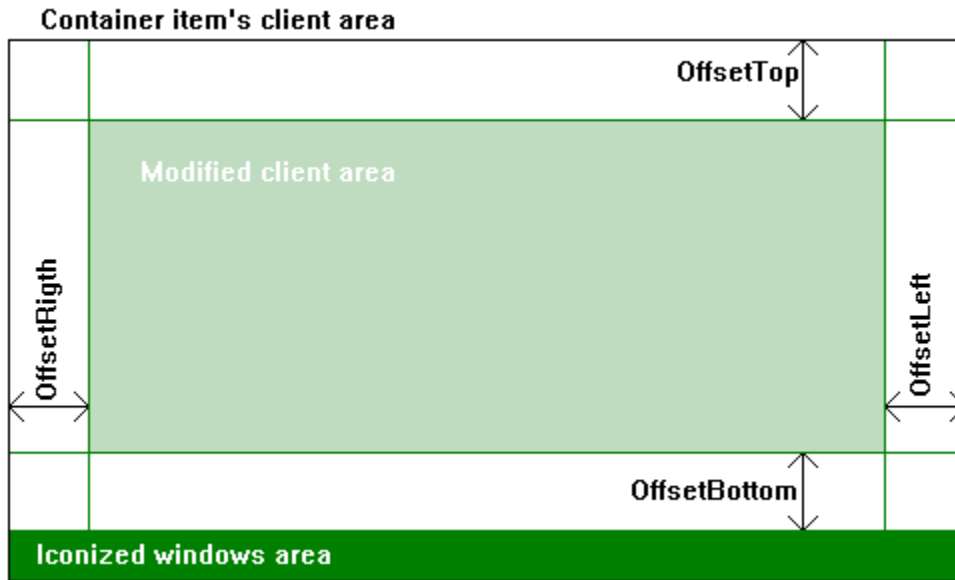
The key name.

Default:

"Key"

Data type:

String



Form & Control Sizer - Version 1.0

IniVariableSectionName Property

Object:

Determines the section name of a variable to be written or readed from the intialization file.

Setting:

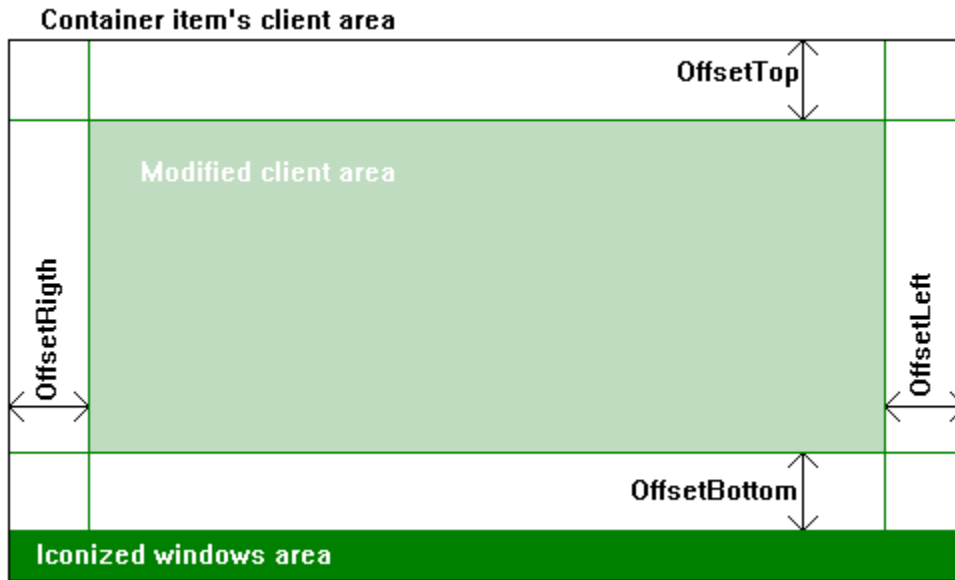
The section name.

Default:

"Section"

Data type:

String



Form & Control Sizer - Version 1.0

IniVariableStringValue Property

Object:

Determines a variable string value to be written or readed from the intialization file.

Setting:

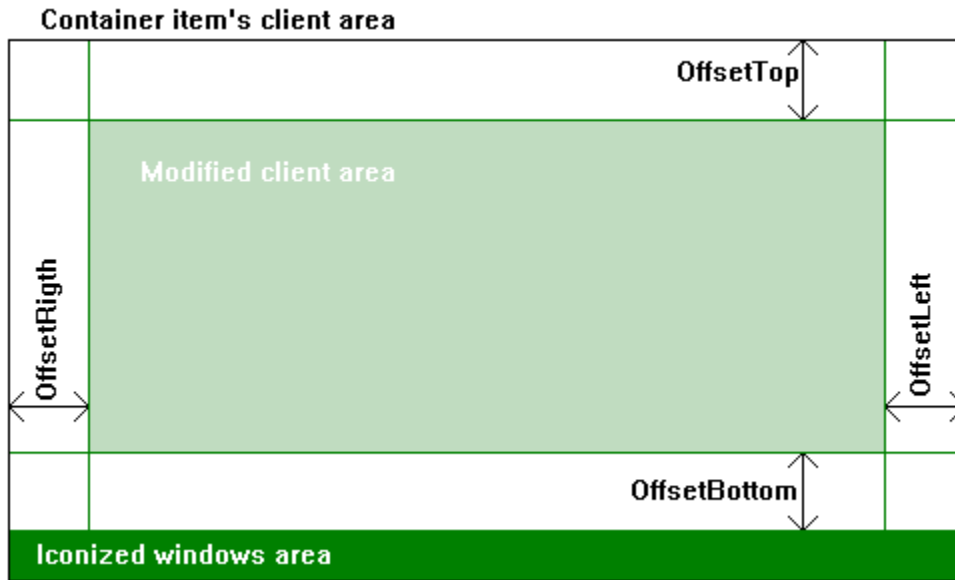
The string value.

Default:

""

Data type:

String



Form & Control Sizer - Version 1.0

OffsetBottom Property

Object:

Determines the offset at the bottom of the container item client area.

Setting:

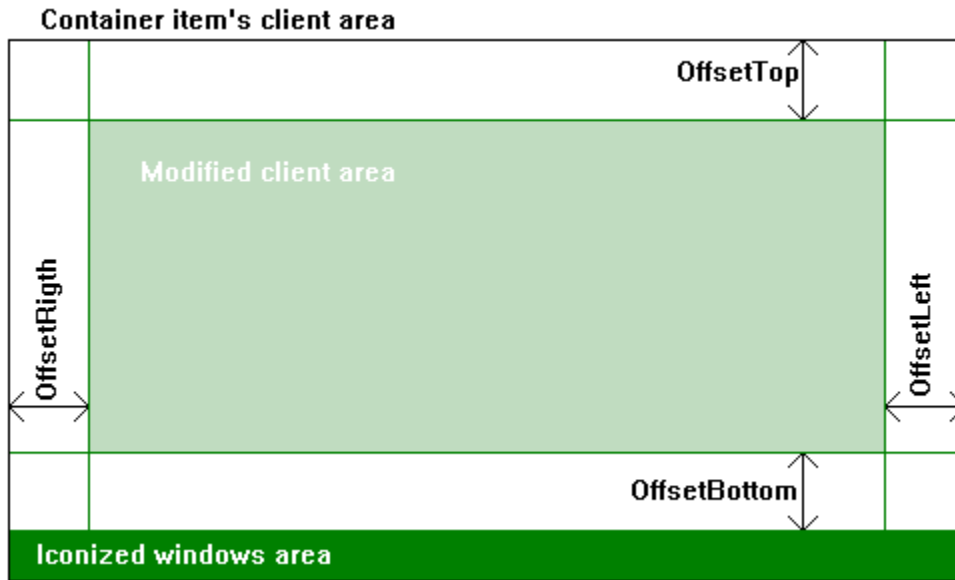
The offset value in pixels.

Default:

0

Data type:

Integer



Form & Control Sizer - Version 1.0

OffsetLeft Property

Object:

Determines the offset at the left of the container item client area.

Setting:

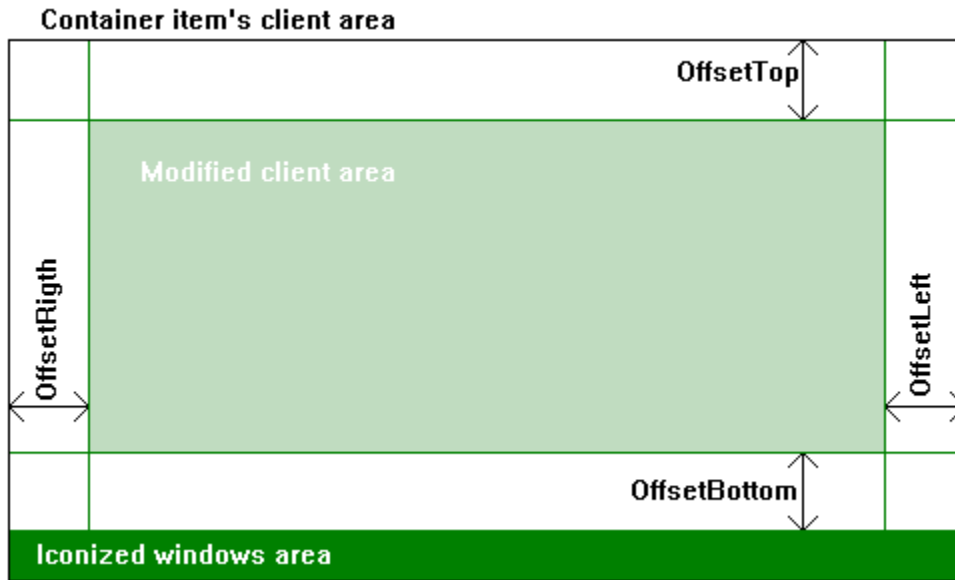
The offset value in pixels.

Default:

0

Data type:

Integer



Form & Control Sizer - Version 1.0

OffsetRight Property

Object:

Determines the offset at the right of the container item client area.

Setting:

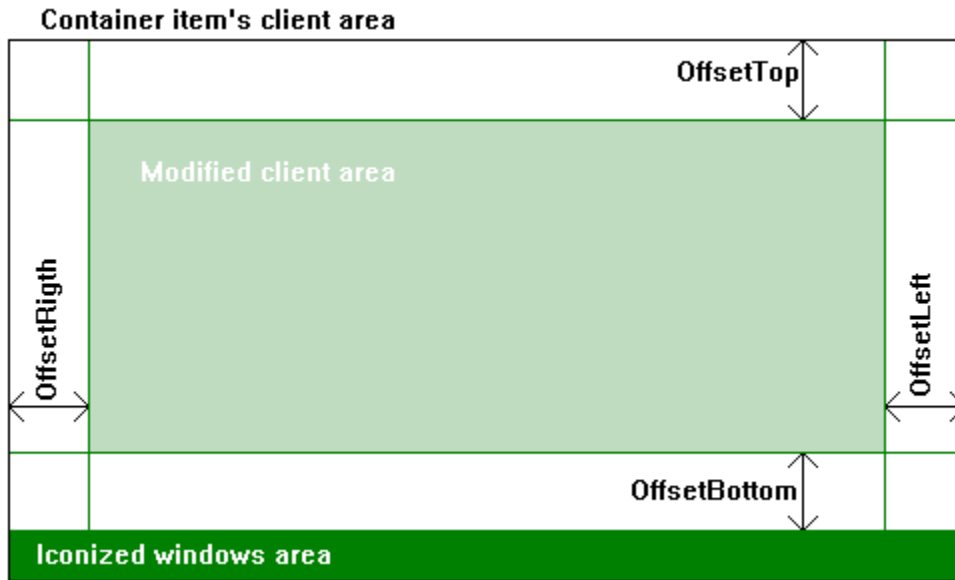
The offset value in pixels.

Default:

0

Data type:

Integer



Form & Control Sizer - Version 1.0

OffsetTop Property

Object:

Determines the offset at the top of the container item client area.

Setting:

The offset value in pixels.

Default:

0

Data type:

Integer

**I can be contacted at my E-Mail address:
lombartm@mail.interpac.be.**

**You may also read my web page where you will
found the last version of this software. Informations
about new products are also available. The address
of this page is:
<http://www.wp.com/SaintRochTree>.**

**If the E-Mail or Web Page addresses change in the
future, you will find the newest at the most Internet
white pages.**

Initialization files are divided in Section. In each section data are stored by key names. They look as:

[SECTION]

Key = XYZ

..... more keys

..... more sections

