Licensing Information

THB Componentware grants a license to use the enclosed software to the original purchaser. Copies may be made for back-up purposes only. Copies made for any other purpose are expressly prohibited, and adherence to this requirement is the sole responsibility of the purchaser.
Customer written executable applications containing embedded THB Componentware products may be freely distributed, without royalty payments to THB Componentware, provided that such distributed THB Componentware product is bound into these applications in such a way so as to prohibit separate use in design mode, and that such THB Componentware product is distributed only in conjunction with the customers own software product. The THB Componentware product may not be distributed by itself in any form.

Neither source code for THB Componentware products nor modified source code for THB Componentware products may be distributed under any circumstances, nor may you distribute .OBJ, .LIB, etc. files that contain our routines. This control may be used as a constituent control only if the compound control thus created is distributed with and as an integral part of an application. Permission to use this control as a constituent control does not grant a right to distribute the license (LIC) file or any other file other than the control executable itself. This license may be transferred to a third party only if all existing copies of the software and its documentation are also transferred.

This product is licensed for use by only one developer at a time. THB Componentware expressly prohibits installing this product on more than one computer if there is any chance that both copies will be used simultaneously. This restriction also extends to installation on a network server, if more than one workstation will be accessing the product. All developers working on a project which includes a THB Componentware product, even though not working directly with the THB Componentware product, are required to purchase a license for that THB Componentware product.

This software is provided as is. THB Componentware makes no warranty, expressed or implied, with regard to the software. All implied warranties, including the warranties of merchantability and fitness for a particular use, are hereby excluded.
THB COMPONENTWARE‘S LIABILITY IS LIMITED TO THE PURCHASE PRICE. Under no circumstances shall THB Componentware or the authors of this product be liable for any incidental or consequential damages, nor for any damages in excess of the original purchase price.

Contact Information

THB Componentware
Herzog Helmut
Hatzenbach 3
A - 2011 Sierndorf
Europe Austria

Web: http://www.thb.findhere.com

E-Mail: office@thb.findhere.com

THBResize Methods

Properties Methods Events THBResize

AddResizeSpecial(ResizeSpecial As THBResizeSpecial)

Adds a control that requires special positioning.
THBResizeSpecial

AddResizeSpecialDirect(ControlName As String, hWnd As Long, FixLeft As thbFixHoriz, FixTop As thbFixVert, FixRight As thbFixHoriz, FixBottom As thbFixVert)

Adds a control that requires special positioning.(just another way)

StartResize()

Only necessary if frames are used. Call this function for each Resizer in the BeforeStartResize event of the form.

THBResize Properties

Properties Methods Events THBResize

HideControlsSmallerThanWidth As Double

Returns or sets the width for hiding controls. If a control gets smaller than this it becomes invisible.

Note: If you work with invisible controls set this property to 0 or your invisible controls will become visible.

HideControlsSmallerThanHeight As Double

Returns or sets the height for hiding controls. If a control gets smaller than this it becomes invisible.

Note: If you work with invisible controls set this property to 0 or your invisible controls will become visible.

HWnd As Long

Returns or sets theWindow Handle of the window to operate on. Without a valid value for this property the control can’t do anything,. So don’t forget to set this property.
Usually you use the Form_Load event to pass the hWnd of the Form.

```
Private Sub Form_Load()
    THBResize1.HWnd = Me.hWnd
End Sub
```

LimitFormWidthMin As Double

Returns or sets the minimal form width. If the user tries to resize the form out of the limit then he will not be able to do this.

LimitFormWidthMax As Double

Returns or sets the maximal form width. If the user tries to resize the form out of the limit then he will not be able to do this.

LimitFormHeightMin As Double

Returns or sets the minimal form height. If the user tries to resize the form out of the limit then he will not be able to do this.

LimitFormHeightMax As Double

Returns or sets the maximal form height. If the user tries to resize the form out of the limit then he will not be able to do this.

NoFontResizeAll As Boolean

Returns or sets if the fonts are resized too. You can set this property for each control extra in the THBResizeSpecial object.

ResizeOnlyDirectChilds As Boolean

Returns or sets if the the childs of child windows are resized too.
If there are frames on a form and these frames contain controls then you can explicit decide if these controls within the frames are resized too.

FontSizeRelativeToWidthAndHeight As Boolean

Returns or sets if the the font size is calculated relative to form height or relative to width and height. The default value is True.

THBResizeSpecial Properties

Properties Methods Events THBResizeSpecial

ControlName As String

Returns or sets the name of the control these settings belong to. One of the properties ControlName or HWnd must be set to identify the control.

HWnd As Long

Returns or sets the window handle of the control these settings belong to. One of the properties ControlName or HWnd must be set to identify the control.

FixCompleteToTop As Boolean

Fix this control relative to top window border.

FixCompleteToBottom As Boolean

Fix this control relative to bottom window border.

FixCompleteToLeft As Boolean

Fix this control relative to left window border.

FixCompleteToRight As Boolean

Fix this control relative to right window border.

FixTop As thbFixVert

Returns or sets how to fix the controls top border.

Valid values:

Name	Value	Meaning
thbFixVertNone	0	don‘t fix
thbFixVertTop	1	fix to top
thbFixVertBottom	2	fitx to bottom

FixBottom As thbFixVert

Returns or sets how to fix the controls bottom border.

Valid values:

Name	Value	Meaning
thbFixVertNone	0	don‘t fix
thbFixVertTop	1	fix to top
thbFixVertBottom	2	fitx to bottom

FixLeft As thbFixHoriz

Returns or sets how to fix the controls left border.

Valid values:

Name	Value	Meaning
thbFixHorizNone	0	don‘t fix
thbFixHorizLeft	1	fix to left
thbFixHorizRight	2	fitx to right

FixRight As thbFixHoriz

Returns or sets how to fix the controls right border.

Valid values:

Name	Value	Meaning
thbFixHorizNone	0	don‘t fix
thbFixHorizLeft	1	fix to left
thbFixHorizRight	2	fitx to right

KeepWidth As Boolean

Returns or sets if the control keeps its width.

KeepHeight As Boolean

Returns or sets if the control keeps its height.

KeepAspect As Boolean

Returns or sets if the control keeps its proportion.

NoFontResize As Boolean

Returns or sets to deactivate font resizing. If True then the font keeps its original size.

DeactivateAutoResize As Boolean

Returns or sets to deactivate resizing for this control.

THBResize Control

Properties Methods Events THBResize

Your job is to write sophisticated applications doing great things. So don’t mess around with positioning controls on the form. Delegate this job to THBResize and concentrate on more important things.

Implemented within minutes.
Extendable to all your needs.

Fast guide:

- Download the control and the example code
- Start a new VB 6 project
- Add some controls
- Add THBResize Control to the form
- Add the following lines of code:

```
Private Sub Form_Load()
     THBResize.HWnd = Me.hWnd
End Sub
```

- That’s it. Have fun resizing your form...

To get some inspiration have a look at the example code.

THBResize resizes all controls on the form whenever the user resizes the form. So the form gets elastic and offers a perfect overall impression no matter how it is sized.
As soon as you place the control on the form and add 3 lines of code all controls behave elastic.
If you have special whishes to some controls you can either define these with a **visual User Interface** contained in the property pages of the THBResize control or code it by hand.

Tip:
For controls that you resize it’s better to use TrueType fonts like ‚Arial‘ or ‚Times New Roman‘ because they look good regardless of their size.

Supported ActiveX Control Containers:
VB 6,
VC 6 (container doesn’t support visual editing) Please read the special VC notes.

THBResizeSpecial Object

Properties Methods Events

Use this COM Object to define the resizing behaviour for one control.
To identify the control specify ControlName or HWnd.
Add the COM Object to the THBResize Control by calling AddResizeSpecial.

Redistributing THBResize

Before an ActiveX control can be used in your distributed application, it must be included in the Windows Registry. You register an ActiveX Control with the tool regsvr32.exe. But normally this is done by your setup wizard, so you don’t have to struggle around with this.

Required Files:
To redistribute THBResize, you must include the following files to your setup package:

File	**Description**	**Version**	**Register**
OLEPRO32.DLL	Placeholder for old DLL	5.0.4275	Yes
OLEAUT32.DLL	Central automation DLL	2.40.4275 or greater	Yes
THBResXX.dll	ActiveX Control		Yes

Note: OLEAUT32.DLL and OLEPRO32.DLL are included in the VB 6 Runtime files. So they will be installed automatically. Nevertheless you must ensure that the version is up to date. If you do not do so, your installation might fail or produce unexpected results.

You can download these files in the Dependencies section of our homepage.

VB5:

If you are using Visual Basic 5 setup wizard to create your installation, check that the DLL files the wizard incorporates are up to date. The setup wizard pulls DLL files for the installation it creates from the \SETUPKIT\KITFIL32\SYS32 directory. If those DLL files are not up to date, you must replace them.

VB6:

The Package& Deployment wizard pulls the files from your Windows System Directory.

THBResize Events

Properties Methods Events THBResize

BeforeStartResize
BeforeResize
AfterResize

BeforeStartResize()

This event is necessary if there are more Resizers on the form. You need more than one Resizer if you work with frames.
Use the event of the outermost THBResize control. The outermost THBResize control is the one that includes all frames. The BeforeStartResize events of all other THBResize controls will not be needed.
Within this event call StartResize of all THBResize controls.

A typical BeforeStartResize event may look like this:

```
Private Sub THBResize1_BeforeStartResize()
    THBResize1.StartResize
    THBResize2.StartResize
    THBResize3.StartResize
    THBResize4.StartResize
End Sub
```

BeforeResize(ContainerWidthOrig As Long, ContainerWidthNew As Long, ContainerWidthPerc As Double, ContainerHeightOrig As Long, ContainerHeightNew As Long, ContainerHeightPerc As Double)

Is fired before the first control on the form is resized. You can use this event to manually add some resize behavior.
The passed values provide you the current form size and the original form size.
Note: If you work with frames and the first control on the form is within a frame you get the size from the frame and not the size of the form.

AfterResize(ContainerWidthOrig As Long, ContainerWidthNew As Long, ContainerWidthPerc As Double, ContainerHeightOrig As Long, ContainerHeightNew As Long, ContainerHeightPerc As Double)

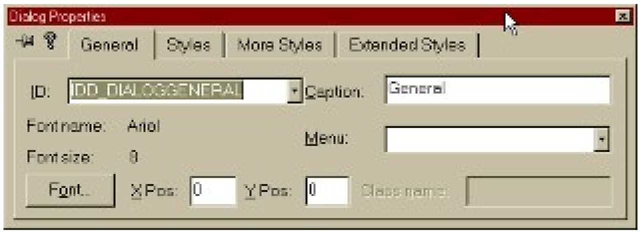
Is fired after the last control on the form is resized. You can use this event to manually add some resize behavior.
The passed values provide you the current form size and the original form size.

Note: If you work with frames and the first control on the form is within a frame you get the size from the frame and not the size of the form.

Special Notes for Visual C++ Users

If you use THBResize within Visual C++ these tips might help you to create more professional dialogs.

1.
If you insert a new dialog to your project, go to the Dialog Properties and change the font to a TrueType font.



ID: IDD_DIALOGGENERAL
Caption: General
Font name: Arial
Menu:
Font size: 8
X Pos: 0
Y Pos: 0
Class name:

For controls that you resize it’s better to use TrueType fonts like ‚Arial‘ or ‚Times New Roman‘ because they look good regardless of their size.

2.
On the ‚Styles‘ Tab of the Dialog Properties‘ dialog select ClipChildren. This ensures that there is no flickering when you resize your dialog.
But be careful, as mentioned in the VC Help, you should use ClipChildren only if there are no group boxes on your dialog.
Thats only partially true, if you set the Transparent Style of all your group boxes, they will be displayed correct, but they will flicker.

3.
The ‚Special Resize Settings‘ Tab of THBResize Property Pages doesn’t work with the control container provided in VC. But everything can be done manually.