

Known Limitations

The ReSize custom control is able to resize most controls with top, left, height and width properties. ReSize can not operate on text that has been printed on a form or in a picture box. The ReSize control may actually end up clearing this text as a result of a resize operation.

In some cases there is a limit to the extent that ReSize can shrink a font. This is a Windows restriction, not one imposed by ReSize. (For more information see [resizing fonts with the ReSize custom control](#)).

ReSize has been tested with the standard controls that ship with Visual Basic™ Professional Edition, but it may or may not work with any given 3rd party controls. Some grid and tab controls will not work properly with ReSize.

Resizing fonts with the ReSize custom control

The ReSize custom control will resize the fonts of all of the controls on a form as well as resizing the controls themselves.

Be aware that some fonts will not shrink beyond a certain point and that some fonts will substitute Small Fonts when they shrink beyond a certain point. In the case of fonts that substitute Small Fonts, ReSize is smart enough to see that the original font is restored to each control when the control again assumes a size that is large enough to support the originally specified font.

While most controls are able to change size in pixel increments, fonts are scaled according to point size. Fonts do not always scale in perfect proportion to the controls that contain them.

Properties

<u>About</u>	Displays an about box.
Left	Stores the position of control.
Name	Contains the name of the ReSize control.
Tag	User property.
Top	Stores the position of the control.
<u>Version</u>	Contains version information for the control.

Events

The ReSize control has no events.

Methods

The ReSize control has no methods. It takes action automatically when the form that contains it is resized.

About Property

Clicking the about property from the property box in the Visual Basic™ development environment or setting the about property will bring up the ReSize about box.

Version Property

The version property contains the numeric version of the ReSize control. This property also contains the word Shareware, Runtime, or Registered depending on which version of the control you are running.

Release Notes for version 2.00

The line control is now properly resized:

The line control that ships with Visual Basic™ is now properly handled by ReSize. The line control is a special case, because it does not contain the standard left, top, height, and width properties. The line control will now resize properly even when you set the ScaleWidth and ScaleHeight properties of a form to custom values.

Support for the MicroHelp tab control:

ReSize will now work correctly with the MicroHelp tab control that ships with VBTools 4.

Fonts are now limited at 1.9 point size:

Some fonts appear to have trouble when ReSize sets their point size to less than 1.9, but only in certain circumstances. These fonts include Arial, Arial Rounded Bold, Britanic Bold, and Times Roman New. These are fonts that switch to Small Fonts when sized below about 6 points. Since 1.9 point seems as small as anyone would ever want to set a font, we have elected to limit fonts from being set smaller than this. Doing so corrects the problem. We choose not to try to predict which fonts will have the problem and under what circumstances. We think our solution is faster more compact and perhaps more reliable than predicting exactly when the problem will occur.

Minimizing a form is handled better:

When a form is minimized, ReSize no longer alters the size of the controls on the form. This makes applications run a little faster and avoids some problems that previously could cause ReSize to generate a stack overflow.

A memory leak in ReSize 1.50 has been corrected:

ReSize 1.50 could cause Visual Basic™ to display an out of memory error. This was due to a memory leak in the 1.50 version of ReSize. This has now been corrected.

.FRX files are now included in distributed files:

.FRX files that make up part of the VBSAMPLE program were omitted in the past. This has now been corrected.

See also: [Release Notes for version 1.50](#)

Release Notes for version 1.50

Controls can now be moved at runtime:

ReSize version 1.50 is smart enough to keep track of controls that are moved around and sized by VB code at runtime. Version 1.00 would simply move these controls back to their original relative positions and sizes each time the form was resized.

Fonts can now be changed at runtime:

ReSize version 1.50 now allows you to change all font properties (i.e. fontname, fontsize, etc.) at runtime. Version 1.00 would set changed fonts back to their design time property values each time the form was resized.

Control now contains a version stamp:

ReSize now contains version information that can be read by ver.dll. This is helpful when you include resize in projects that are redistributed with the Visual Basic™ setup wizard and other installation tools.

License bug fixed:

The registered version of ReSize 1.00 would not load into the development environment unless the ReSize.LIC file was in the Visual Basic™ working directory. The documentation directed registered developers to place the ReSize.LIC file in the \windows directory. This has been corrected and ReSize will now function correctly in the development environment if the ReSize.LIC file is in either the Visual Basic™ working directory or the \windows directory.

Text not staying bold bug:

ReSize 1.00 would unbold text in some cases when controls were resized to very small sizes and then made large again. This is now corrected.

Features and Tips

[Using ReSize with grid controls](#)

[Moving controls around on the form](#)

[Retrofitting existing applications](#)

[If a frame or other container doesnt repaint properly after resizing a form](#)

Retrofitting existing applications

Retrofitting existing applications is as easy as placing a ReSize control on each form you wish to give resizing capabilities.

Repaint problems with Frames

We have seen some cases where ReSize has trouble repainting the background of some container controls after a form resize. This will be fixed in a future version of ReSize. The present workaround is to place a statement of the form `FRAME1.Refresh` in the resize event of the form where you are experiencing the problem.

Moving controls around on the form

If you move controls around on a form under control of your program, ReSize is smart enough to detect this. The controls will be positioned and sized in accordance with the new positions assigned by your Visual Basic™ program.

Using ReSize with grid controls

ReSize is able to resize the outer dimensions of most grid controls, but it does not resize each individual cell of the grid. In some cases, the cells of a grid may appear to be changing in size due to changes in the font size which ReSize DOES make to the grid control as it is resized.

There are many grid controls available and they all handle sizing of cells a little bit differently. At least at this point, we have not attempted to make ReSize recognize each brand of grid control and handle its resizing requirements as a special case.

It is fairly easy to combine ReSize with some of your own Visual Basic™ code to correctly resize the cells of a grid control. The following code handles resizing of cells in the grid control that ships with Visual Basic™. You should be able to write similar code for other grid controls on the market.

Create a form with a ReSize control and a grid control. Set the grid to 5 columns and 9 rows. Place the following code in the form resize event. In practice you would probably use a for loop.

```
Sub Form_Resize ()
    Grid1.ColWidth(0) = Grid1.Width / 5
    Grid1.ColWidth(1) = Grid1.Width / 5
    Grid1.ColWidth(2) = Grid1.Width / 5
    Grid1.ColWidth(3) = Grid1.Width / 5
    Grid1.ColWidth(4) = Grid1.Width / 5
    Grid1.RowHeight(0) = Grid1.Height / 9
    Grid1.RowHeight(1) = Grid1.Height / 9
    Grid1.RowHeight(2) = Grid1.Height / 9
    Grid1.RowHeight(3) = Grid1.Height / 9
    Grid1.RowHeight(4) = Grid1.Height / 9
    Grid1.RowHeight(5) = Grid1.Height / 9
    Grid1.RowHeight(6) = Grid1.Height / 9
    Grid1.RowHeight(7) = Grid1.Height / 9
    Grid1.RowHeight(8) = Grid1.Height / 9
End Sub
```

Please note that you can divide by numbers other than 9 and 5 to get varied effects. Note also that you don't have to divide each row or column by the same number.

The above code forces cell sizes to be relative to the total size of the grid control which ReSize is manipulating. This works because ReSize changes the sizes and locations of the controls on the form before the form resize event fires.

ReSize Custom Control

[Properties](#)

[Events](#)

[Methods](#)

[Known Limitations](#)

[Register](#)

[Release notes](#)

[Features & Tips](#)

Description

A ReSize control resizes all other controls on the same form each time the form changes its size.

File Name

RESIZE.VBX

Object Type

ReSize

Remarks

Place a ReSize control anywhere on a form and it will cause the other controls on the form to resize along with the form. This saves you from writing code. ReSize is so simple to use that you don't need to set even a single property to make it work. ReSize is also smart enough to resize fonts on most controls. (See [resizing fonts with the ReSize custom control](#) for more information).

You must place a ReSize control on each form that you want to affect. You should not place two ReSize controls on a single form.

The shareware version of the ReSize control will only operate in the development environment. To find out how to get a registered version of ReSize, see the [register](#) help topic.

Registering ReSize

When you register your copy of ReSize, you will receive the most recent version of ReSize. The registered version of ReSize will run in both the runtime and development Visual Basic™ environments. The shareware version of ReSize runs only in the development environment. The nag screen is not present in the registered version. Registered users of ReSize may distribute the ReSize runtime for free.

To register your version of ReSize, refer to #5614 in the CompuServe SWREG forum or send a check or money order for \$29.00 to:

Larcom and Young
P.O. Box 66379
Roseville, Michigan 48066-6379

(Canada and Mexico add \$3.00 all others outside the U.S. add \$5.00 shipping and handling)

Please include:

Your Name

Your Company Name (Optional)

Mailing Address

Your City, State ZIP

Phone Number where you can be reached

