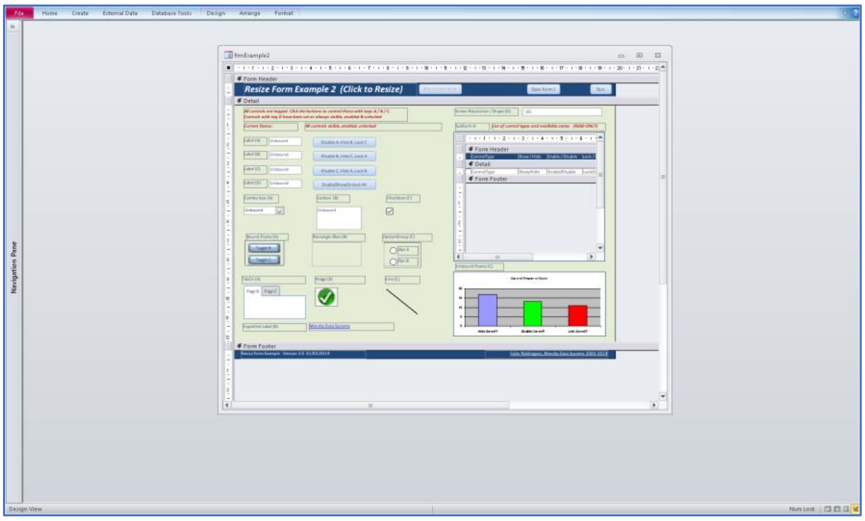
# *ResizeForm Me – A Tutorial in Automatic Form Resizing*

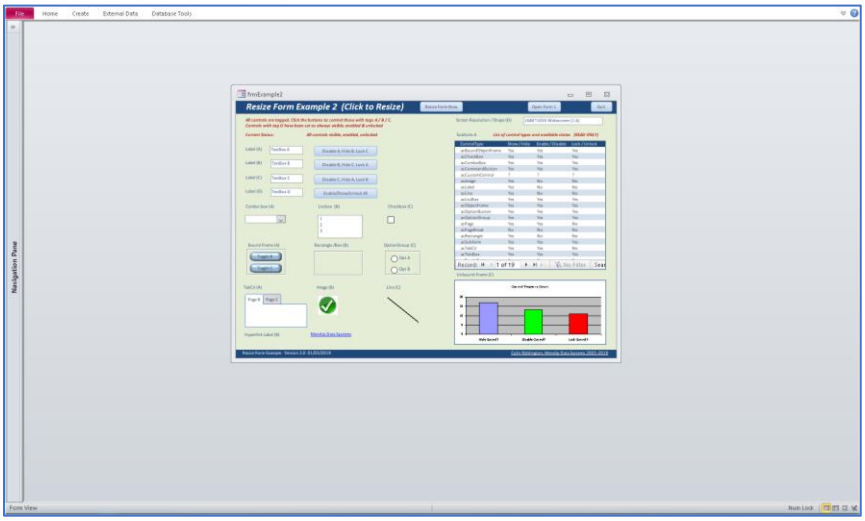
## *Version 2.6 – Updated 14/03/2019*

All developers will be aware that forms designed for a specific screen size/resolution may look dreadful on a different monitor with a higher/lower resolution and /or screen size or shape (4:3, 16:9 or widescreen)

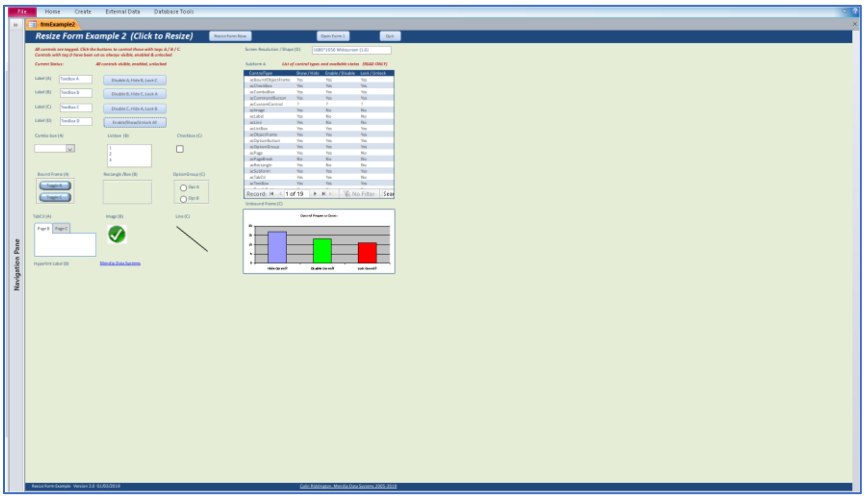
For example, this form was designed at a low resolution **(800*600)**



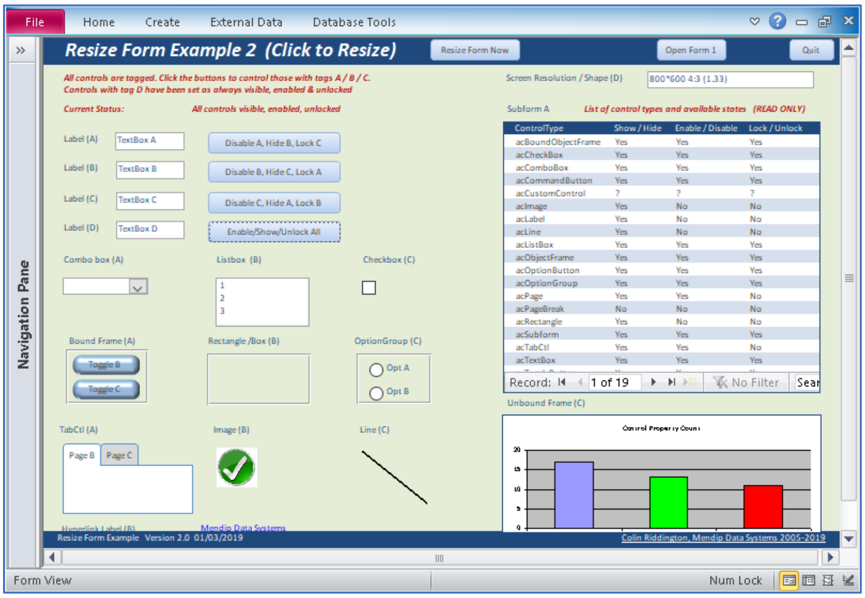
If it is viewed at a higher resolution **(1680*1020)**, it only fills part of the screen and each item is tiny
This is using overlapping windows



Or with **tabbed documents**, the screen is filled but the contents remain squashed in the top left corner



On a smaller screen or lower resolution, the opposite problem occurs with part of the form not shown unless the user scrolls in both directions



All controls are tagged. Click the buttons to control those with tags A / B / C.
Controls with tag D have been set as always visible, enabled & unlocked

Current Status: All controls visible, enabled, unlocked

Screen Resolution / Shape (D) 800*600 4:3 (1.33)

Subform A List of control types and available states (READ ONLY)

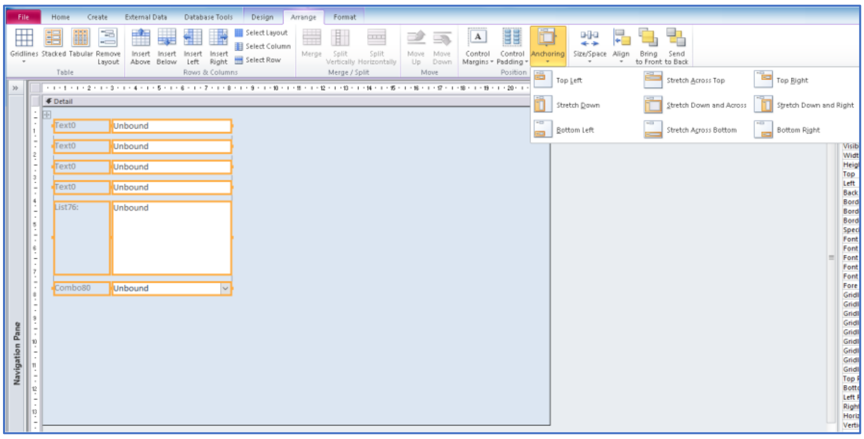
| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |

Resize Form Example  Version 2.0  01/03/2019

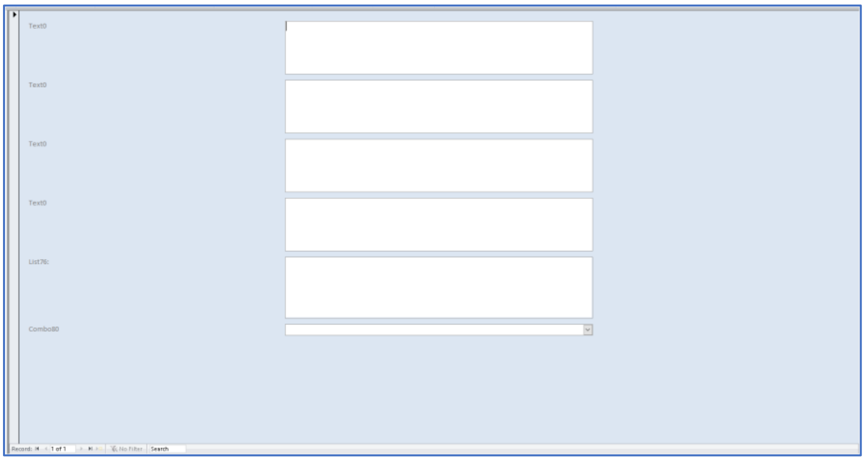
Colin Riddington, Mendip Data Systems 2005-2019

The effect is the same whether using overlapping windows or tabbed documents

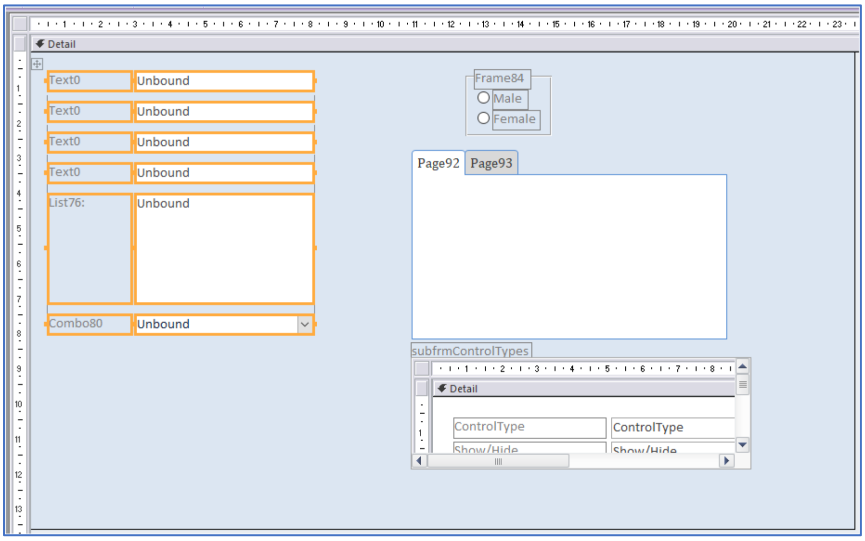
Some developers get around this issue to some extent by using layout guides to group several controls together.
For example, a stacked layout has been used for these controls.
All controls in the group are automatically made the same width



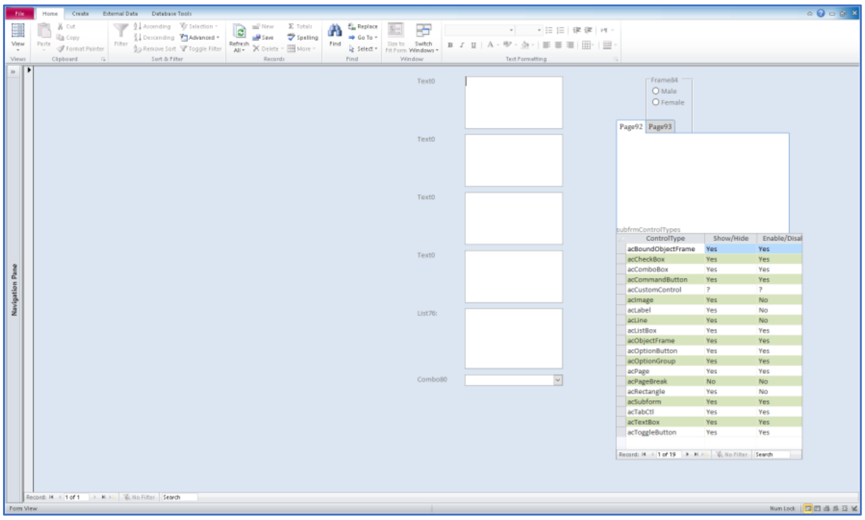
In addition, anchoring can be used to adjust the position/size on the screen.
In this case **Stretch Down and Across** has been applied



The results can be quite successful for simple form layouts. However, that isn't necessarily the case for more complex form designs. For example, extra controls have been added in design view:



Depending on the anchoring choices, this may result in this (with lots of empty space on the left)

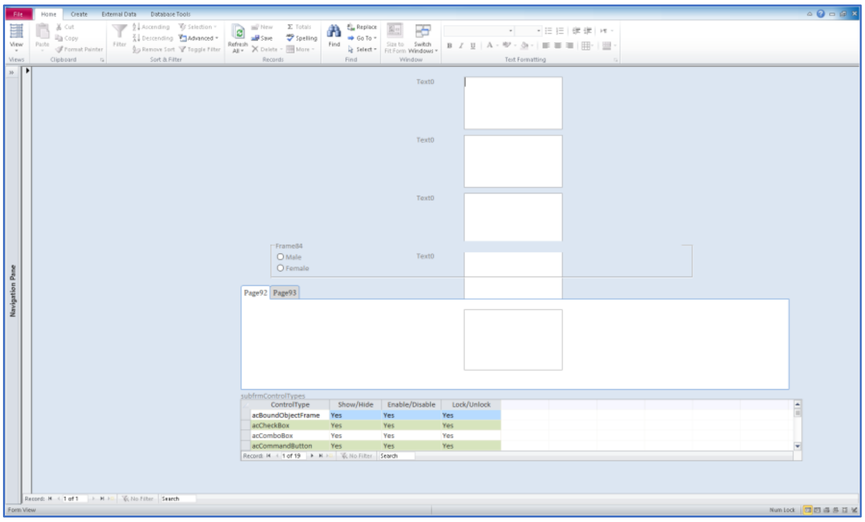


subfrmControlTypes

| ControlType | Show/Hide | Enable/Disal |
|---|---|---|
| acBoundObjectFrame | Yes | Yes |
| acCheckBox | Yes | Yes |
| acComboBox | Yes | Yes |
| acCommandButton | Yes | Yes |
| acCustomControl | ? | ? |
| acImage | Yes | No |
| acLabel | Yes | No |
| acLine | Yes | No |
| acListBox | Yes | Yes |
| acObjectFrame | Yes | Yes |
| acOptionButton | Yes | Yes |
| acOptionGroup | Yes | Yes |
| acPage | Yes | Yes |
| acPageBreak | No | No |
| acRectangle | Yes | No |
| acSubform | Yes | Yes |
| acTabCtl | Yes | Yes |
| acTextBox | Yes | Yes |
| acToggleButton | Yes | Yes |

Record: 1 of 19

Or another variation with various controls overlapping / obscured … etc



subfrmControlTypes

| ControlType | Show/Hide | Enable/Disable | Lock/Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |

Record: 1 of 19

With planning, reasonable results can still be obtained.
However, users do not have full control over positioning and size of each control on the form

**Form resizing code** is designed to fix all such issues. Forms can be **automatically resized** for any **screen size & resolution** whilst allowing developers full control over the form layout and appearance

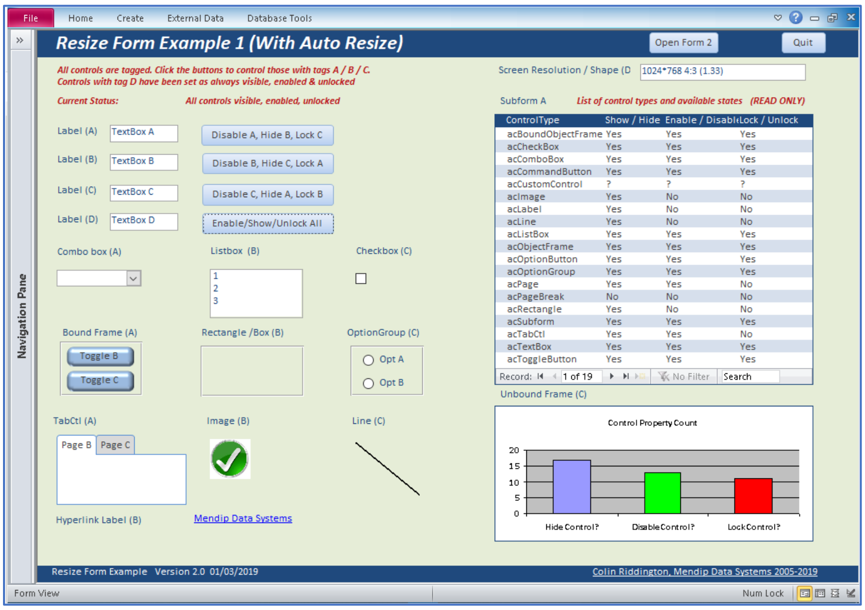
There are various examples available including commercial packages such as:

- **Shrinker Stretcher** available from http://www.peterssoftware.com/ss.htm
- **Total Access Components** from http://www.fmsinc.com/microsoftaccess/controls.html

However, for many years, I have used a modified version of **free open source code** by **Jamie Czernik** originally written in 2003. The original code is still available at http://jamieczernik.powweb.com/articles/resolution.html though various improvements have been added since by others including myself.

This is an **identical form** to that shown earlier in this article but with form resizing code added.

Here is the form viewed at a resolution of **1024*768**



*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

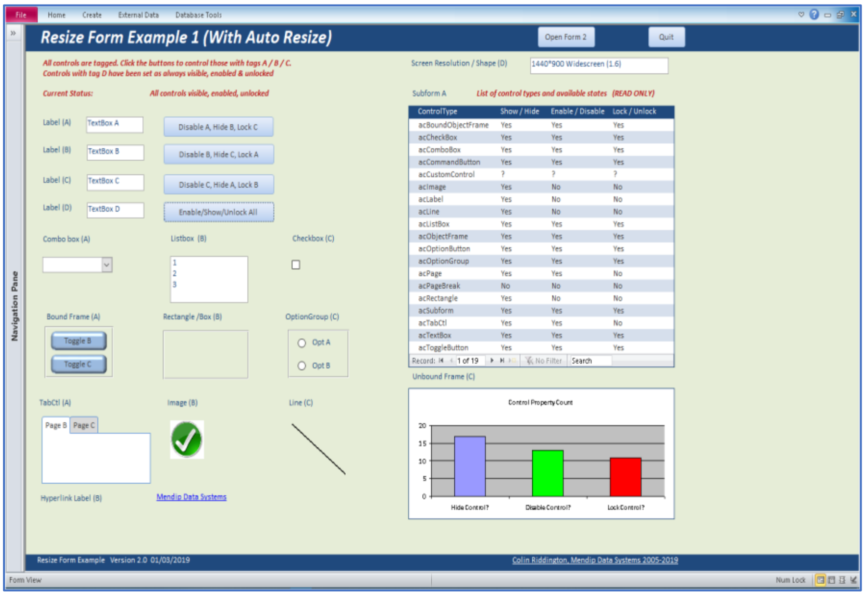
Screen Resolution / Shape (D 1024*768 4:3 (1.33)

*List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disabl | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

Record: 1 of 19 No Filter Search

And again at a resolution of **1440 *900**



*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

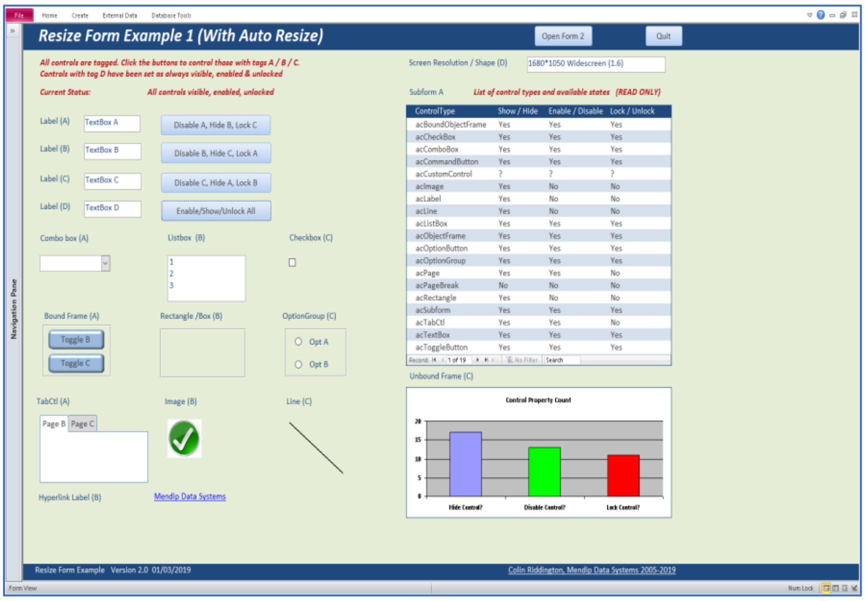
Screen Resolution / Shape (D) 1440*900 Widescreen (1.6)

*List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

Record: 1 of 19 No Filter Search

And yet again at a resolution of **1680*1050**



Resize Form Example 1 (With Auto Resize)

All controls are tagged. Click the buttons to control those with tags A / B / C.
Controls with tag D have been set as always visible, enabled & unlocked

Current Status: All controls visible, enabled, unlocked

Screen Resolution / Shape (D): 1680*1050 Widescreen (1.6)

Subform A — List of control types and available states (READ ONLY)

| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

Record: 1 of 19 No Filter Search

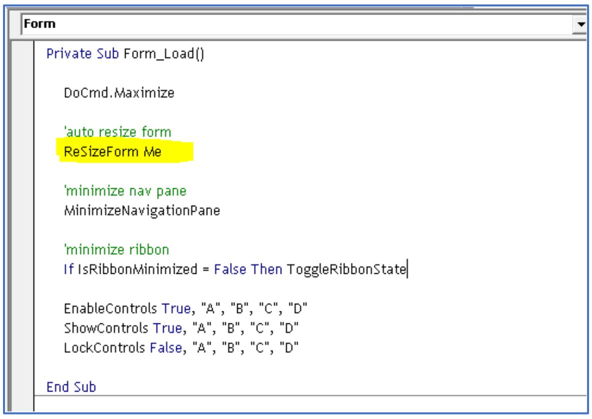
Resize Form Example Version 2.0 01/03/2019

Colin Riddington, Mendip Data Systems 2005-2019

As you can see the entire form is shown whatever resolution is used as long as that is the same as or higher than the 'design resolution'.
The code intelligently enlarges the form itself to fit the screen and moves each control by an appropriate amount to keep the form in proportion. It also enlarges each control proportionately together with the contents of that control. The code works for each type of control including subforms

NOTE:
1. To allow for different form factors (screen shapes) i.e. 4:3, 16:9, widescreen some empty space may be left on the right of the screen when using a maximised form as above
2. The code has been extensively tested on a wide range of monitors and form factors such as 22 inch widescreen, 14 inch 16:9, 15 inch 4:3 and a 10 inch 16:9 tablet

So what code is neeed to work this magic? Just **one line** in the **Form_Load** event – **ResizeForm Me**



```
Private Sub Form_Load()

    DoCmd.Maximize

    'auto resize form
    ReSizeForm Me

    'minimize nav pane
    MinimizeNavigationPane

    'minimize ribbon
    If IsRibbonMinimized = False Then ToggleRibbonState

    EnableControls True, "A", "B", "C", "D"
    ShowControls True, "A", "B", "C", "D"
    LockControls False, "A", "B", "C", "D"

End Sub
```

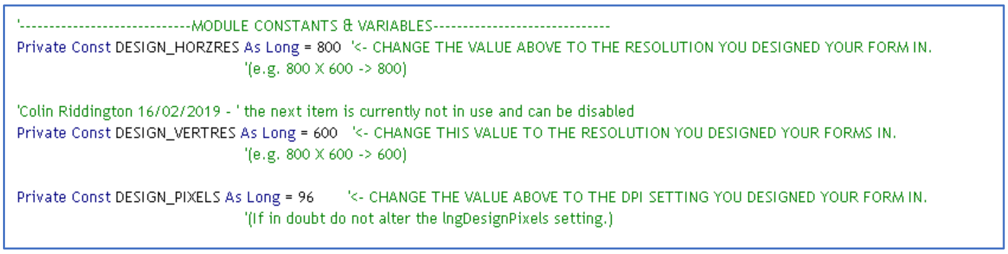
OK, I hear you say, surely that can't be all there is to it! Well not quite ….
Behind the scenes there is a lengthy module **modResizeForm** which contains all the code required to adjust the forms as required. This is included with the attached example application and has been updated to work in both 32-bit and 64-bit Access

First import the module to your own application.
Then add the above line in the **Form_Load** event, the form will be resized.
However, initially the results will not be successful as the forms were not designed with resizing in mind
**It is STRONGLY recommended that you start with new forms and plan for resizing from the start**

### *Designing with the form resizer*

As a general rule, you always need to develop using the lowest resolution that your users are likely to have.
Form resizing upwards (stretching) is much easier and has far fewer issues than resizing downwards (shrinking).

The declarations section of the **modResizeForm** module includes the following lines:



```
'----------------------------MODULE CONSTANTS & VARIABLES----------------------------
Private Const DESIGN_HORZRES As Long = 800  '<- CHANGE THE VALUE ABOVE TO THE RESOLUTION YOU DESIGNED YOUR FORM IN.
                                            '(e.g. 800 X 600 -> 800)

'Colin Riddington 16/02/2019 - ' the next item is currently not in use and can be disabled
Private Const DESIGN_VERTRES As Long = 600  '<- CHANGE THIS VALUE TO THE RESOLUTION YOU DESIGNED YOUR FORMS IN.
                                            '(e.g. 800 X 600 -> 600)

Private Const DESIGN_PIXELS As Long = 96    '<- CHANGE THE VALUE ABOVE TO THE DPI SETTING YOU DESIGNED YOUR FORM IN.
                                            '(If in doubt do not alter the lngDesignPixels setting.)
```

The first line is the default or baseline horizontal resolution – in this case 800
You can adjust this to any other suitable minimum value such as 1024 if you prefer

The next line is the baseline vertical resolution – 600
This can also be amended to suit, but in the current version of the code this is NOT used & can be disabled

The third line is the **pixels per inch** setting which is normally 96 for 100%
NOTE: if the screen is magnified to e.g. 125% this value becomes 120 dpi

### *Setting the default form size*

The code currently specifies a 'base resolution' of ***800*600*** though that can be changed to any resolution that suits your needs. However, I certainly don't develop in that resolution. My primary monitor is in fact ***1680*1050***.

This actually means that my base form size in design view is such that if I did use ***800*600***, it would fill the screen.
By experiment, I found that size should be ***20.5cm*12.5cm*** approximately

Hence my forms are designed in that size & will scale up to 'fill the screen' in any other higher resolution
More accurately it will fill the total height of the screen.

Due to different form factors ***(4:3, 5:4, 16:9, 16:10 etc)*** the form may be slightly less than the total monitor width.
Whilst I could stretch it to fit horizontally, there would be some distortion by doing so
Alternatively, you could increase the default width of your forms in design view

I also design with default font = **Calibri 7pt** which after resizing becomes **Calibri 11pt**

I could change the base resolution to say ***1024*768*** and if I did so, the base form size would be proportionately larger ***(25.85*17.2 cm approx)*** & **font size 9pt**. However, it would scale up from that equally well.

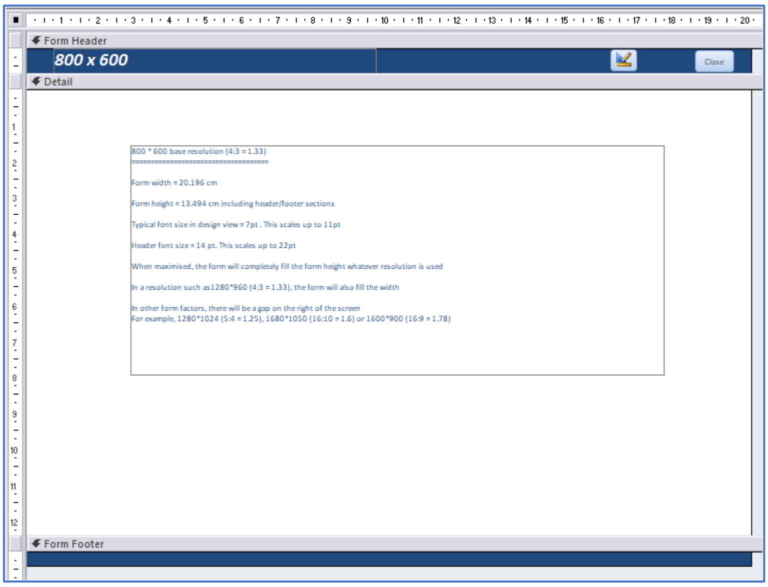
You may wish to adjust the default width if, for example, all users have monitors with widescreen form factor.

You may also need to adapt the specified sizes slightly if you normally do any of the following:

- Set ribbon and / or navigation pane maximised
- Design forms with no header / footer section
- Use tabbed documents instead of overlapping windows
- Use popup forms

The attached application includes two example template forms ***(800x600 & 1024x768)*** which may be useful to set maximum form sizes to ensure they fit properly in different screen resolutions.

### 800*600 template



800 * 600 base resolution (4:3 = 1.33)
====================================

Form width = 20.196 cm

Form height = 13.494 cm including header/footer sections

Typical font size in design view = 7pt . This scales up to 11pt

Header font size = 14 pt. This scales up to 22pt

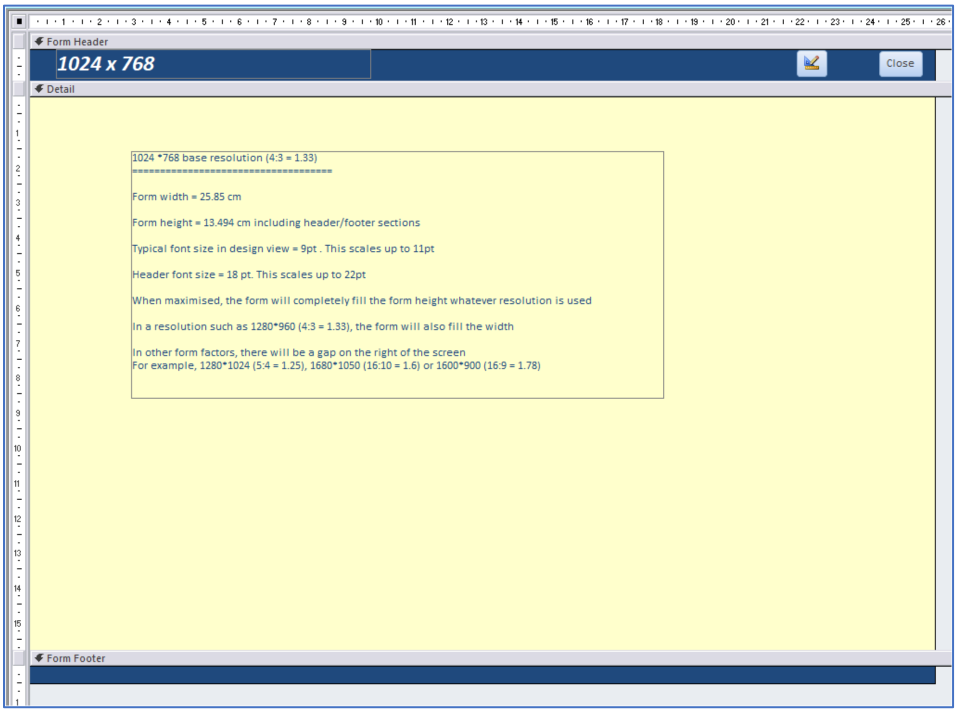
When maximised, the form will completely fill the form height whatever resolution is used

In a resolution such as1280*960 (4:3 = 1.33), the form will also fill the width

In other form factors, there will be a gap on the right of the screen
For example, 1280*1024 (5:4 = 1.25), 1680*1050 (16:10 = 1.6) or 1600*900 (16:9 = 1.78)

NOTE:
The text on the two template forms has been designed to be viewed after resizing when it will be perfectly legible!

### 1024*768 template



1024 *768 base resolution (4:3 = 1.33)
====================================

Form width = 25.85 cm

Form height = 13.494 cm including header/footer sections

Typical font size in design view = 9pt . This scales up to 11pt

Header font size = 18 pt. This scales up to 22pt

When maximised, the form will completely fill the form height whatever resolution is used

In a resolution such as 1280*960 (4:3 = 1.33), the form will also fill the width

In other form factors, there will be a gap on the right of the screen
For example, 1280*1024 (5:4 = 1.25), 1680*1050 (16:10 = 1.6) or 1600*900 (16:9 = 1.78)

**TIP:**
It is useful to place a transparent box control over each section of the form to mark out the correct size required. The box should be equal to the width and height of that section.

This will be useful if you need to restore the original size after making any temporary changes in form layout / size

### *Using the example applications*

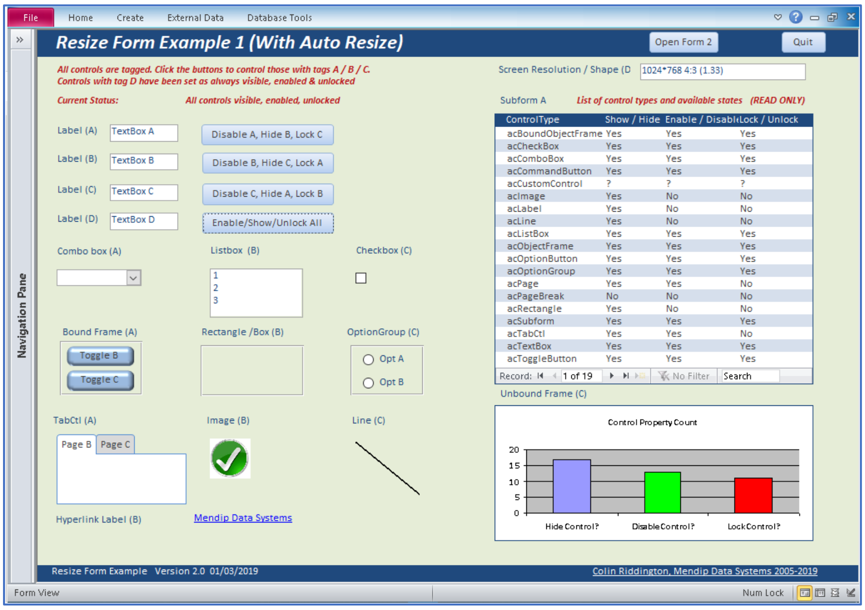
The attached application **ResizeFormExample_v2.6.accdb** has been designed to show how the code works.

**Earlier versions** of the resizing code had issues when using **tabbed documents**.
That issue has been solved in this **latest version (2.6)**. See the **Issues and Solutions** section for details

The **latest version** works equally well with **overlapping windows** or **tabbed documents** display options.

It contains THREE almost identical versions of the main form.
Each form also includes a subform and a wide variety of other controls to show how each is affected by resizing

- **frmExample1** – automatically resized on form load



Resize Form Example 1 (With Auto Resize)

*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

Screen Resolution / Shape (D 1024*768 4:3 (1.33)

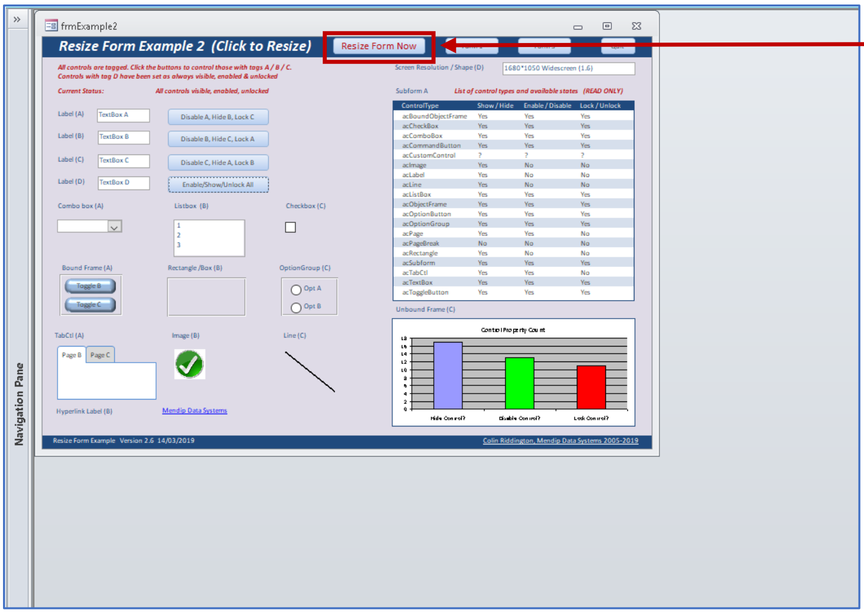
Subform A *List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disabl | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

Record: 1 of 19 No Filter Search

Resize Form Example Version 2.0 01/03/2019 Colin Riddington, Mendip Data Systems 2005-2019

- **frmExample2** – not resized on form load BUT can be resized by clicking a button in the form header



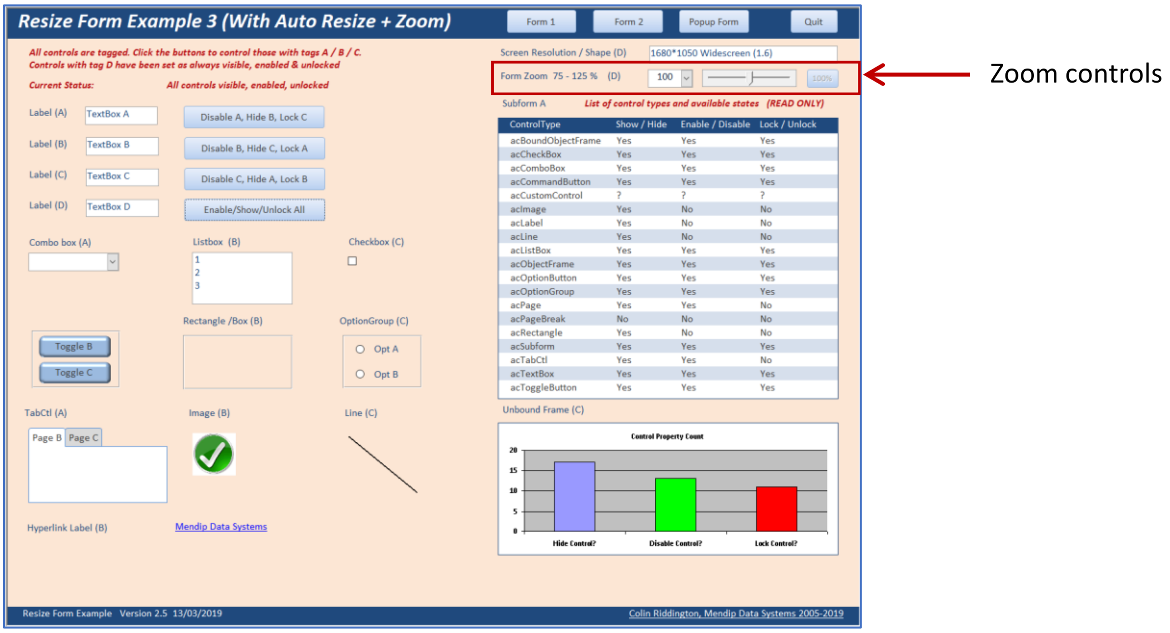
Click to resize

The resize button is automatically disabled once it has been clicked.
This is to prevent the resizing code being applied again.

- **frmExample3** – same as form **frmExample1** but with an additional **zoom feature**

Unlike **Word** and **Excel**, there is no built-in **zoom control** available in **Access**.
For those with less than perfect eyesight this can be an issue.

However, a modified version of the resizing code can also be used to allow users to **zoom forms in/out**
It allows further adjustment of the form from **75% to 125%** of the **default size** for your screen & resolution



**Resize Form Example 3 (With Auto Resize + Zoom)**

*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

TextBox A
TextBox B
TextBox C
TextBox D

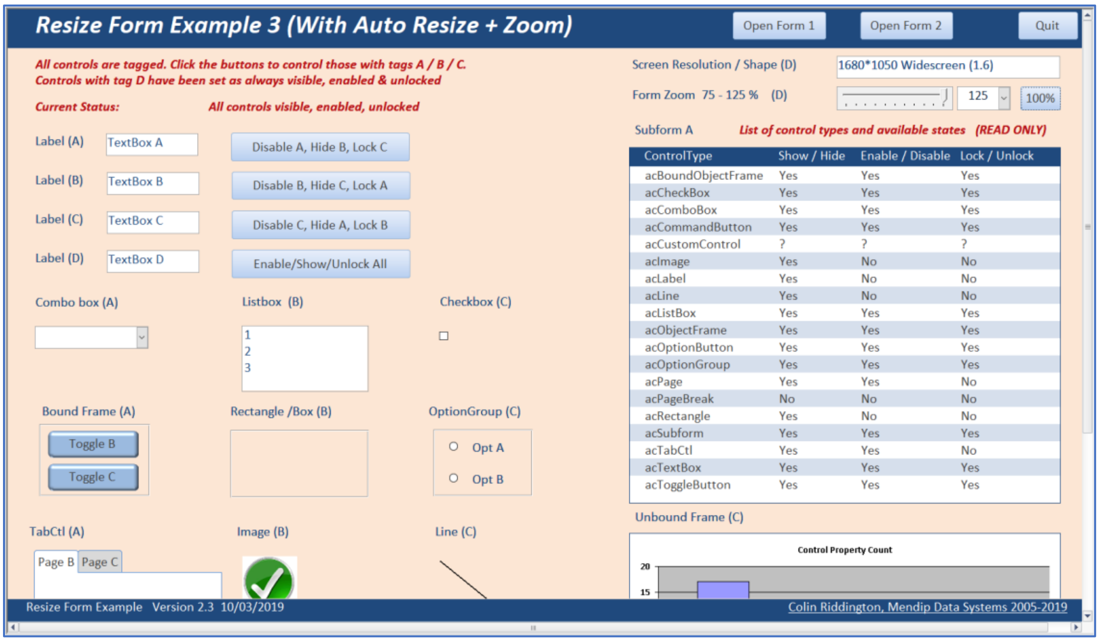
Screen Resolution / Shape (D): 1680*1050 Widescreen (1.6)
Form Zoom 75 - 125 % (D): 100

Subform A *List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

Resize Form Example Version 2.5 13/03/2019
Colin Riddington, Mendip Data Systems 2005-2019

Use the combo or slider control to adjust the scale in 5% intervals according to personal preference.
The form dimensions and all controls SHOULD be adjusted accordingly
For example, with zoom = 125%



**Resize Form Example 3 (With Auto Resize + Zoom)**

*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

TextBox A
TextBox B
TextBox C
TextBox D

Screen Resolution / Shape (D): 1680*1050 Widescreen (1.6)
Form Zoom 75 - 125 % (D): 125

Subform A *List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

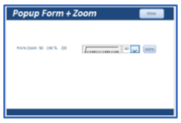
Resize Form Example Version 2.3 10/03/2019
Colin Riddington, Mendip Data Systems 2005-2019

The zoom can be reset to normal by clicking the **100%** button

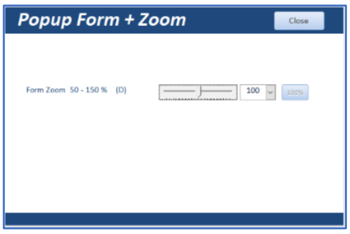
NOTE:
1. Increasing the zoom may cause part of the form to move ‘off screen’ as in the screenshot above. Scrollbars appear automatically when this happens
2. The slider is an **ActiveX** control. This will be hidden and ‘inactive’ if you have disabled ActiveX controls in **Access Options Trust Center**

A very simple **popup form** is also available which includes a **zoom feature** with a range **50 – 150%**

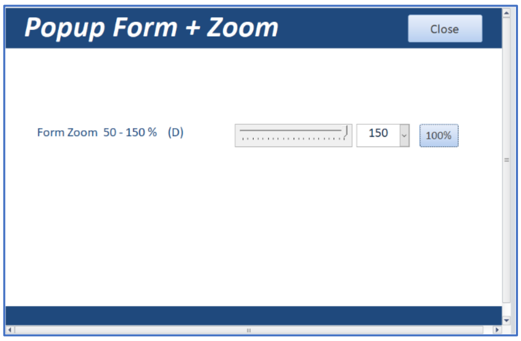
**50%**



**100%**



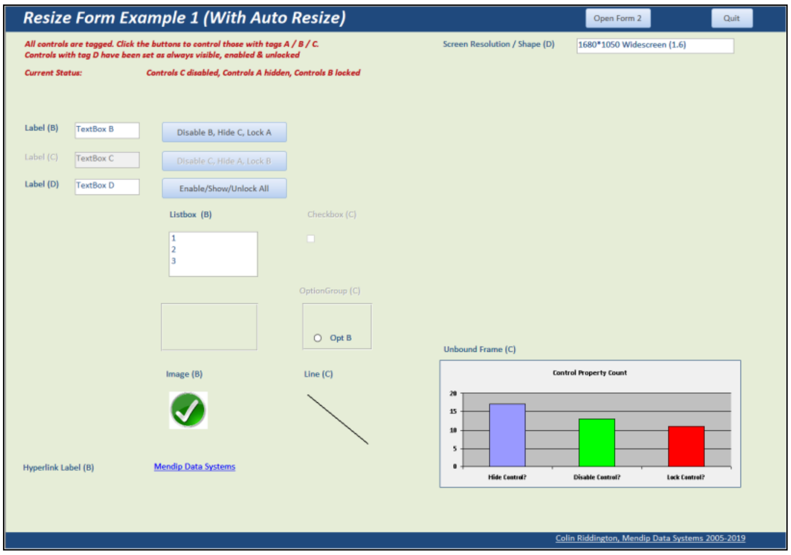
**150%**



Once again, click the 100% button to reset the form to its default size and scale

**NOTE:** The main form used in this example was originally designed to demonstrate the use of the **Tag** property to make **groups** of **controls visible/hidden, enabled/disabled, locked/unlocked**.

In this screenshot, controls with tag A have been hidden, tag B controls locked and tag C controls disabled.



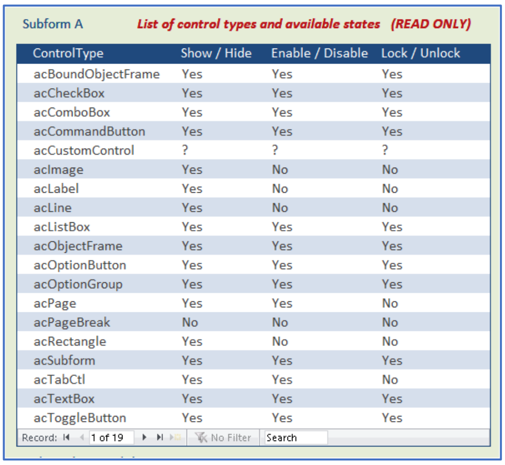
*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *Controls C disabled, Controls A hidden, Controls B locked*

Screen Resolution / Shape (D): 1680*1050 Widescreen (1.6)

All the code used in that feature is contained in the module **modControlState**
For more details, see [http://www.mendipdatasystems.co.uk/set-controls/4594398114](http://www.mendipdatasystems.co.uk/set-controls/4594398114)

NOTE: Not all controls can be hidden / disabled / locked.
The subform summarises the properties available for each control type

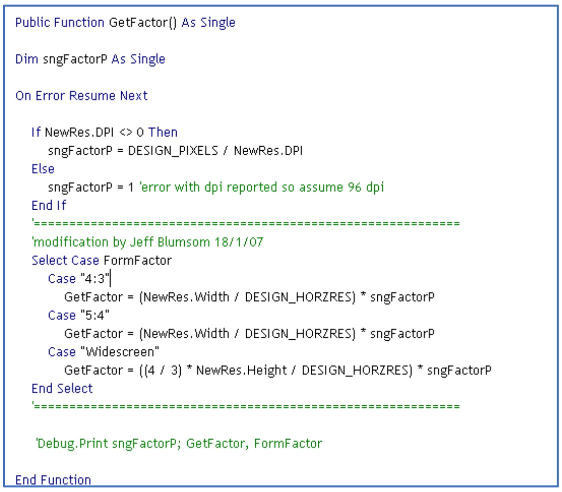

Subform A
Record: 1 of 19
No Filter
Search

*List of control types and available states (READ ONLY)*

| ControlType | Show / Hide | Enable / Disable | Lock / Unlock |
|---|---|---|---|
| acBoundObjectFrame | Yes | Yes | Yes |
| acCheckBox | Yes | Yes | Yes |
| acComboBox | Yes | Yes | Yes |
| acCommandButton | Yes | Yes | Yes |
| acCustomControl | ? | ? | ? |
| acImage | Yes | No | No |
| acLabel | Yes | No | No |
| acLine | Yes | No | No |
| acListBox | Yes | Yes | Yes |
| acObjectFrame | Yes | Yes | Yes |
| acOptionButton | Yes | Yes | Yes |
| acOptionGroup | Yes | Yes | Yes |
| acPage | Yes | Yes | No |
| acPageBreak | No | No | No |
| acRectangle | Yes | No | No |
| acSubform | Yes | Yes | Yes |
| acTabCtl | Yes | Yes | No |
| acTextBox | Yes | Yes | Yes |
| acToggleButton | Yes | Yes | Yes |

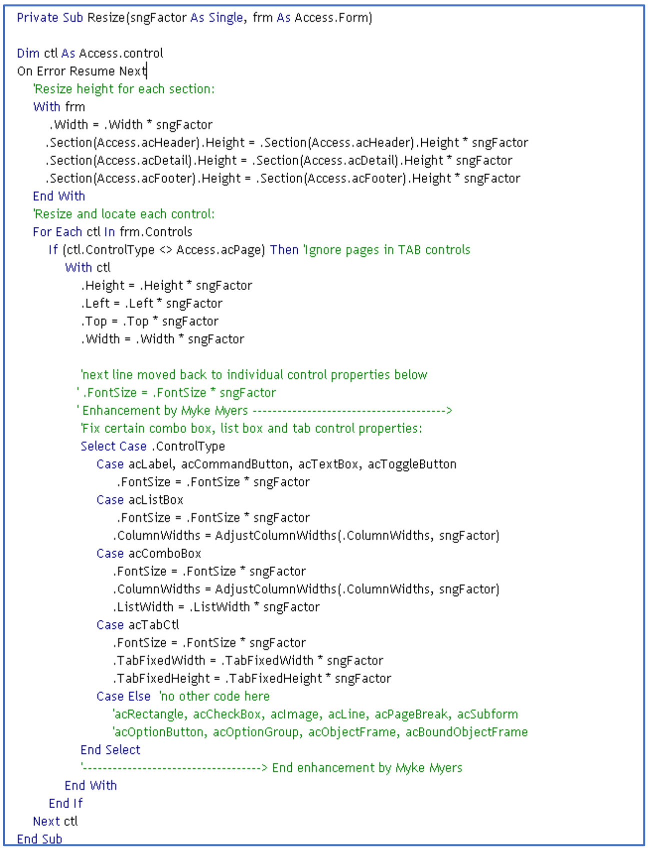
## *How Does the Code Work?*

The **GetFactor** function calculates the multiplying factor based on screen size & resolution:



```
Public Function GetFactor() As Single

Dim sngFactorP As Single

On Error Resume Next

    If NewRes.DPI <> 0 Then
        sngFactorP = DESIGN_PIXELS / NewRes.DPI
    Else
        sngFactorP = 1 'error with dpi reported so assume 96 dpi
    End If
    '=============================================================
    'modification by Jeff Blumsom 18/1/07
    Select Case FormFactor
        Case "4:3"
            GetFactor = (NewRes.Width / DESIGN_HORZRES) * sngFactorP
        Case "5:4"
            GetFactor = (NewRes.Width / DESIGN_HORZRES) * sngFactorP
        Case "Widescreen"
            GetFactor = ((4 / 3) * NewRes.Height / DESIGN_HORZRES) * sngFactorP
    End Select
    '=============================================================


     'Debug.Print sngFactorP; GetFactor, FormFactor

End Function
```

The **Resize** procedure then uses that information to adjust the height and width of the form together with the size and position of each control on the form



```
Private Sub Resize(sngFactor As Single, frm As Access.Form)

Dim ctl As Access.control
On Error Resume Next
    'Resize height for each section:
    With frm
        .Width = .Width * sngFactor
       .Section(Access.acHeader).Height = .Section(Access.acHeader).Height * sngFactor
       .Section(Access.acDetail).Height = .Section(Access.acDetail).Height * sngFactor
       .Section(Access.acFooter).Height = .Section(Access.acFooter).Height * sngFactor
    End With
    'Resize and locate each control:
    For Each ctl In frm.Controls
        If (ctl.ControlType <> Access.acPage) Then 'Ignore pages in TAB controls
            With ctl
                .Height = .Height * sngFactor
                .Left = .Left * sngFactor
                .Top = .Top * sngFactor
                .Width = .Width * sngFactor

                'next line moved back to individual control properties below
               ' .FontSize = .FontSize * sngFactor
               ' Enhancement by Myke Myers -------------------------------------->
                'Fix certain combo box, list box and tab control properties:
                Select Case .ControlType
                    Case acLabel, acCommandButton, acTextBox, acToggleButton
                         .FontSize = .FontSize * sngFactor
                    Case acListBox
                         .FontSize = .FontSize * sngFactor
                        .ColumnWidths = AdjustColumnWidths(.ColumnWidths, sngFactor)
                    Case acComboBox
                        .FontSize = .FontSize * sngFactor
                        .ColumnWidths = AdjustColumnWidths(.ColumnWidths, sngFactor)
                        .ListWidth = .ListWidth * sngFactor
                    Case acTabCtl
                        .FontSize = .FontSize * sngFactor
                        .TabFixedWidth = .TabFixedWidth * sngFactor
                        .TabFixedHeight = .TabFixedHeight * sngFactor
                    Case Else  'no other code here
                        'acRectangle, acCheckBox, acImage, acLine, acPageBreak, acSubform
                        'acOptionButton, acOptionGroup, acObjectFrame, acBoundObjectFrame
                End Select
                '---------------------------------> End enhancement by Myke Myers
            End With
        End If
    Next ctl
End Sub
```

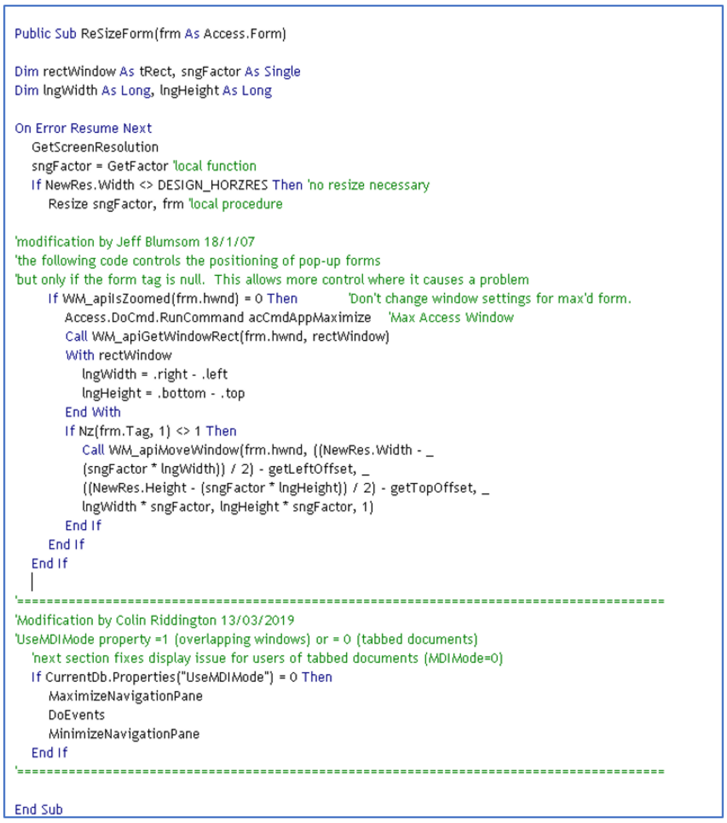
Certain controls receive 'special treatment' – **list boxes, combo boxes and tab controls**
**Tab control pages** are excluded as are the **contents** of **subforms** (the subform **container** is resized automatically)

As previously stated, the code line **ResizeForm Me** needs to be added to each form being resized.
This code line means the **ResizeForm** procedure is applied to the loaded form (**Me**)

To scale up the **subform control positions / sizes**, add the line **ResizeForm Me** to the **Form_Load** event of the **subform**. Alternatively, add a line like **ReSizeForm subFormName.Form** to the **Form_Load** event of the **main form**

**NOTE: DO NOT DO BOTH METHODS** or the **subform** will be **scaled up twice!!**

The main **ResizeForm** code first checks whether resizing is required then uses the **Resize** procedure above

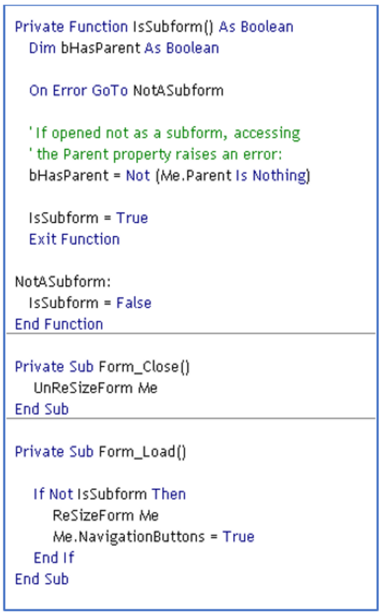


```
Public Sub ReSizeForm(frm As Access.Form)

Dim rectWindow As tRect, sngFactor As Single
Dim lngWidth As Long, lngHeight As Long

On Error Resume Next
   GetScreenResolution
   sngFactor = GetFactor 'local function
   If NewRes.Width <> DESIGN_HORZRES Then 'no resize necessary
      Resize sngFactor, frm 'local procedure

'modification by Jeff Blumsom 18/1/07
'the following code controls the positioning of pop-up forms
'but only if the form tag is null.  This allows more control where it causes a problem
      If WM_apiIsZoomed(frm.hwnd) = 0 Then           'Don't change window settings for max'd form.
         Access.DoCmd.RunCommand acCmdAppMaximize    'Max Access Window
         Call WM_apiGetWindowRect(frm.hwnd, rectWindow)
         With rectWindow
            lngWidth = .right - .left
            lngHeight = .bottom - .top
         End With
         If Nz(frm.Tag, 1) <> 1 Then
            Call WM_apiMoveWindow(frm.hwnd, ((NewRes.Width - _
            (sngFactor * lngWidth)) / 2) - getLeftOffset, _
            ((NewRes.Height - (sngFactor * lngHeight)) / 2) - getTopOffset, _
            lngWidth * sngFactor, lngHeight * sngFactor, 1)
         End If
      End If
   End If
   |
'=========================================================================================
'Modification by Colin Riddington 13/03/2019
'UseMDIMode property =1 (overlapping windows) or = 0 (tabbed documents)
   'next section fixes display issue for users of tabbed documents (MDIMode=0)
   If CurrentDb.Properties("UseMDIMode") = 0 Then
      MaximizeNavigationPane
      DoEvents
      MinimizeNavigationPane
   End If
'=========================================================================================

End Sub
```

Following that two adjustments may be made to fix positioning issues:
a) For **popup forms** an adjustment is made to the code depending on its Tag value.
b) Where the display option is set to tabbed documents, the navigation pane is first maximised and then minimised again.

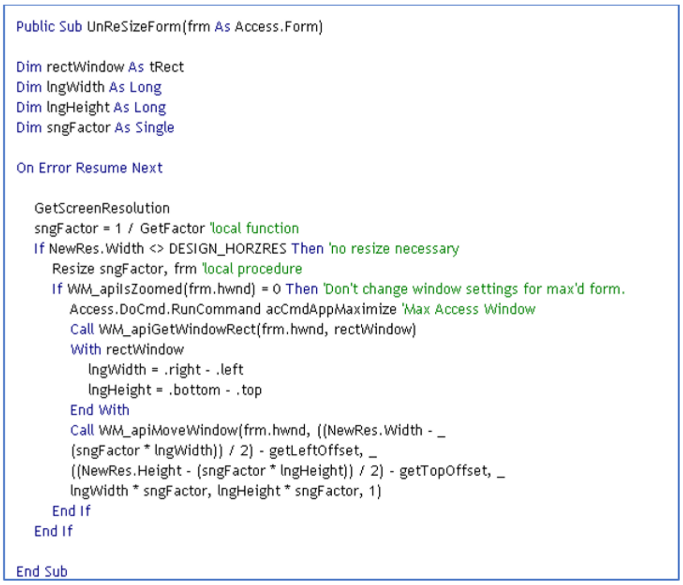
Both adjustments are explained in the **Issues and Solutions** section later in the article

HINT:
If you have a **form** that is sometimes used as a **subform**, you can conditionally apply resizing code when used as a standalone form. Place the following code in the **subform** itself



```
Private Function IsSubform() As Boolean
  Dim bHasParent As Boolean

  On Error GoTo NotASubform

  ' If opened not as a subform, accessing
  ' the Parent property raises an error:
  bHasParent = Not (Me.Parent Is Nothing)

  IsSubform = True
  Exit Function

NotASubform:
  IsSubform = False
End Function

Private Sub Form_Close()
   UnReSizeForm Me
End Sub

Private Sub Form_Load()

   If Not IsSubform Then
      ReSizeForm Me
      Me.NavigationButtons = True
   End If
End Sub
```

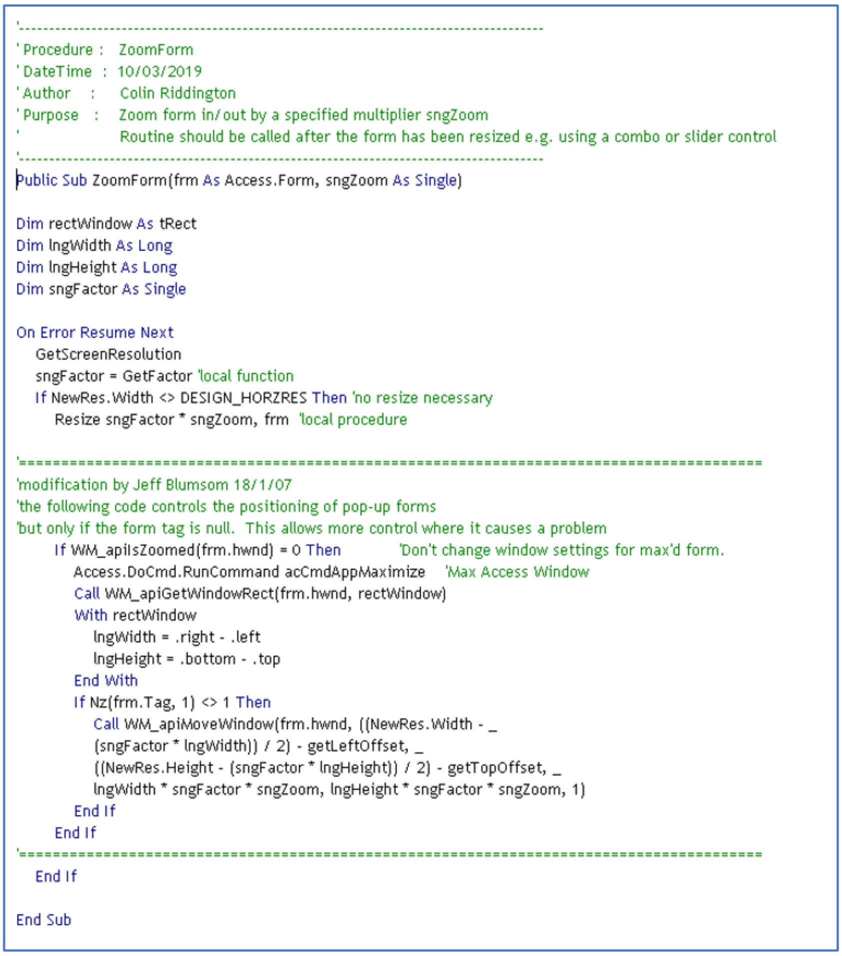
The **UnResizeForm** procedure is used to fix issues if forms become 'over-enlarged'.  See **Issues and Solutions**
This reverses the form resizing code and effectively 'shrinks' the form again.



```
Public Sub UnReSizeForm(frm As Access.Form)

Dim rectWindow As tRect
Dim lngWidth As Long
Dim lngHeight As Long
Dim sngFactor As Single

On Error Resume Next

   GetScreenResolution
   sngFactor = 1 / GetFactor 'local function
   If NewRes.Width <> DESIGN_HORZRES Then 'no resize necessary
      Resize sngFactor, frm 'local procedure
      If WM_apiIsZoomed(frm.hwnd) = 0 Then 'Don't change window settings for max'd form.
         Access.DoCmd.RunCommand acCmdAppMaximize 'Max Access Window
         Call WM_apiGetWindowRect(frm.hwnd, rectWindow)
         With rectWindow
            lngWidth = .right - .left
            lngHeight = .bottom - .top
         End With
         Call WM_apiMoveWindow(frm.hwnd, ((NewRes.Width - _
         (sngFactor * lngWidth)) / 2) - getLeftOffset, _
         ((NewRes.Height - (sngFactor * lngHeight)) / 2) - getTopOffset, _
         lngWidth * sngFactor, lngHeight * sngFactor, 1)
      End If
   End If

End Sub
```

Very occasionally, you may experience a form that does not automatically revert to its correct size on closing.
In such cases, try adding the line **UnresizeForm Me** in the **Form_Close** event.
I also do this where **conditional resizing code** is applied to **subforms** (as above)

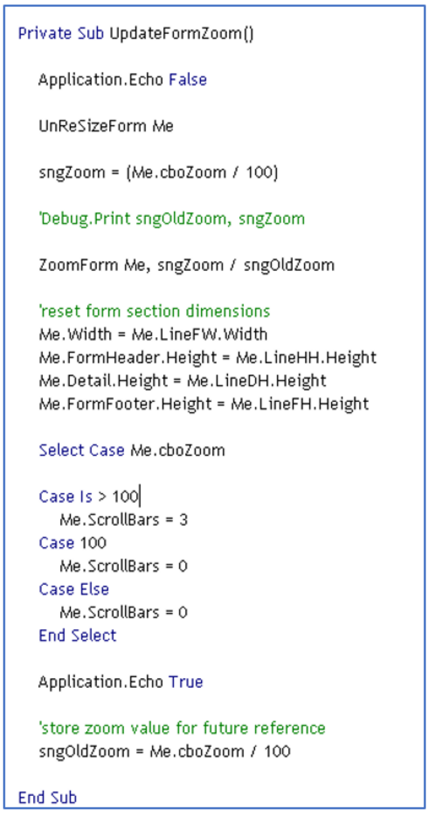
The new **ZoomForm** procedure is a modified version of the oriignal **ResizeForm** code.



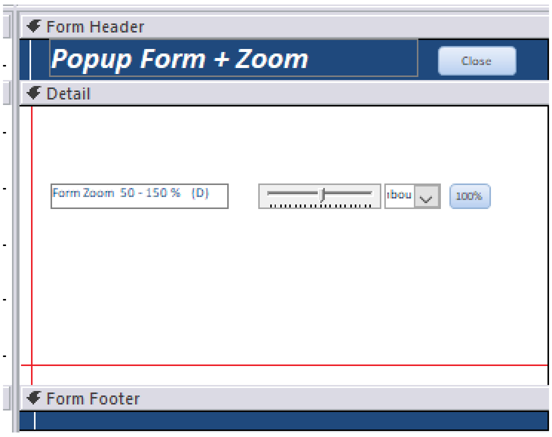
```
'---------------------------------------------------------------------------------------
' Procedure :  ZoomForm
' DateTime  :  10/03/2019
' Author    :  Colin Riddington
' Purpose   :  Zoom form in/out by a specified multiplier sngZoom
'              Routine should be called after the form has been resized e.g. using a combo or slider control
'---------------------------------------------------------------------------------------
Public Sub ZoomForm(frm As Access.Form, sngZoom As Single)

Dim rectWindow As tRect
Dim lngWidth As Long
Dim lngHeight As Long
Dim sngFactor As Single

On Error Resume Next
   GetScreenResolution
   sngFactor = GetFactor 'local function
   If NewRes.Width <> DESIGN_HORZRES Then 'no resize necessary
      Resize sngFactor * sngZoom, frm  'local procedure

'=========================================================================================
'modification by Jeff Blumsom 18/1/07
'the following code controls the positioning of pop-up forms
'but only if the form tag is null.  This allows more control where it causes a problem
      If WM_apiIsZoomed(frm.hwnd) = 0 Then           'Don't change window settings for max'd form.
         Access.DoCmd.RunCommand acCmdAppMaximize    'Max Access Window
         Call WM_apiGetWindowRect(frm.hwnd, rectWindow)
         With rectWindow
            lngWidth = .right - .left
            lngHeight = .bottom - .top
         End With
         If Nz(frm.Tag, 1) <> 1 Then
            Call WM_apiMoveWindow(frm.hwnd, ((NewRes.Width - _
            (sngFactor * lngWidth)) / 2) - getLeftOffset, _
            ((NewRes.Height - (sngFactor * lngHeight)) / 2) - getTopOffset, _
            lngWidth * sngFactor * sngZoom, lngHeight * sngFactor * sngZoom, 1)
         End If
      End If
'=========================================================================================
   End If

End Sub
```

Additional code is used in the form being zoomed.
First of all the form is '**unresized**' then **resized** by an **additional amount** based on the **sngZoom** multiplier

Adjustments are made to the **form width** and **section heights** to ensure it remains in proportion.
This is done by comparing with with the sizes of four hidden lines placed on each section of the form.



```
Private Sub UpdateFormZoom()

    Application.Echo False

    UnReSizeForm Me

    sngZoom = (Me.cboZoom / 100)

    'Debug.Print sngOldZoom, sngZoom

    ZoomForm Me, sngZoom / sngOldZoom

    'reset form section dimensions
    Me.Width = Me.LineFW.Width
    Me.FormHeader.Height = Me.LineHH.Height
    Me.Detail.Height = Me.LineDH.Height
    Me.FormFooter.Height = Me.LineFH.Height

    Select Case Me.cboZoom

    Case Is > 100
        Me.ScrollBars = 3
    Case 100
        Me.ScrollBars = 0
    Case Else
        Me.ScrollBars = 0
    End Select

    Application.Echo True

    'store zoom value for future reference
    sngOldZoom = Me.cboZoom / 100

End Sub
```



Several other procedures are also included in **modResizeForm**:

a) **GetScreenResolution / GetResolution / GetHorizontalResolution / GetVerticalResolution / GetScreenShape**
b) **GetTopOffset / GetLeftOffset** - used to centre forms on screen
c) **AdjustColumnWidths** - used in **Resize** procedure to adjust combo boxes & list boxes during resizing
d) **GetOrigWindow** - used to note the original form dimensions before resizing
e) **RestoreWindow** - can be used in the **Form_Close** event to restore the original form dimensions

## *Issues and Solutions*

The resizing code is most successful with **maximised forms**.
If using **non-maximised forms**, you should check the effect of resizing at different resolutions.
If necessary, adapt the form size/shape for best results.

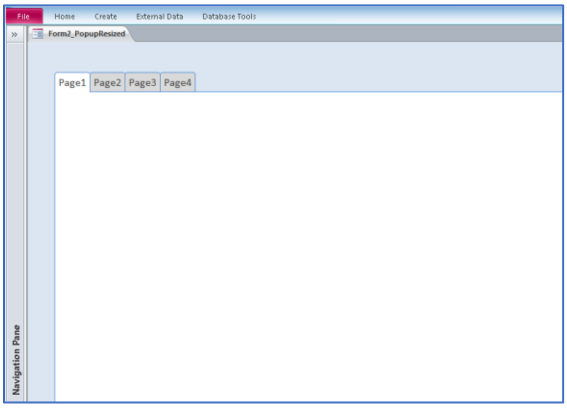
NOTE: Built in **Access forms** like message boxes, input boxes and error messages **are NOT resized**.

**Popup forms** appear on top of the application window and therefore may not fit fully on the screen if resized.
It may be necessary to exclude certain popup forms from being resized.

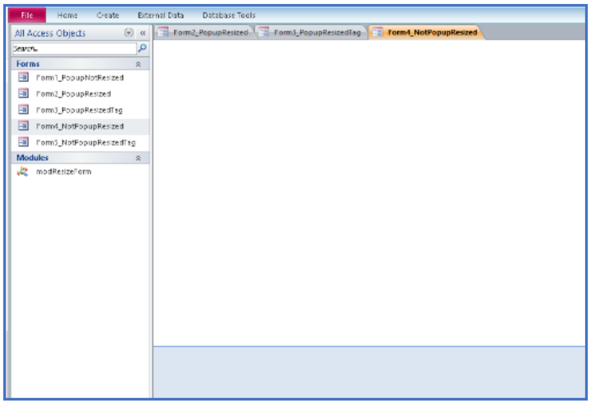
The **original version** of this code by **Jamie Czernik** from 2003 did not always work properly with the **tabbed documents** display option which was first introduced with Access 2007.

For example, certain controls such as **subforms** and **tab controls** were shifted off screen causing the whole form to be **shifted both horizontally and vertically.** For example, using a tab control:

**Tab control in the correct position**



**Tab control shifted down & to the right**



NOTE: the display issue shown above
a) **MAY not occur** with **popup forms** containing **tab controls**
b) does **NOT occur** when using **overlapping windows**

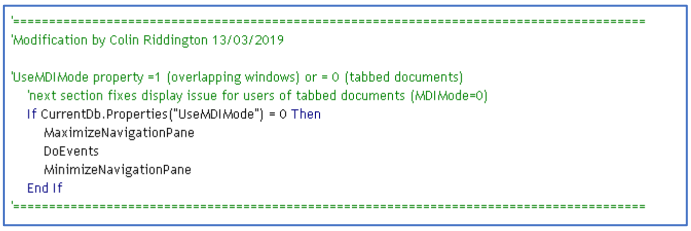
Using both scrollbars, the form can be correctly 'realigned'.
Similarly, if the navigation pane and / or ribbon is maximised then minimised again.
**However even if the form was saved after doing so, it would again be shifted the next time it is opened**

The original partly successful solution was to apply a **tag value = 1 to the form.**
Doing so triggered '**exemption code'** in the **ResizeForm** procedure to be implemented.

However, the **latest version** (v2.5 onwards) of the resizing code supplied with this article **works equally well with both overlapping windows and tabbed documents.**

The code detects the **display option** by checking the **UseMDIMode** property.
The **UseMDIMode** property is 1 for **overlapping windows** or 0 for **tabbed documents**

**After resizing**, the code fix maximizes then minimizes the navigation pane if tabbed documents are in use



```
'===============================================================================
'Modification by Colin Riddington 13/03/2019

'UseMDIMode property =1 (overlapping windows) or = 0 (tabbed documents)
    'next section fixes display issue for users of tabbed documents (MDIMode=0)
    If CurrentDb.Properties("UseMDIMode") = 0 Then
        MaximizeNavigationPane
        DoEvents
        MinimizeNavigationPane
    End If
'===============================================================================
```

Doing this is sufficient to **reset the form positioning** correctly.
It has been tested successfully with **document tabs** both **visible / hidden**.
No fix is required for overlapping windows

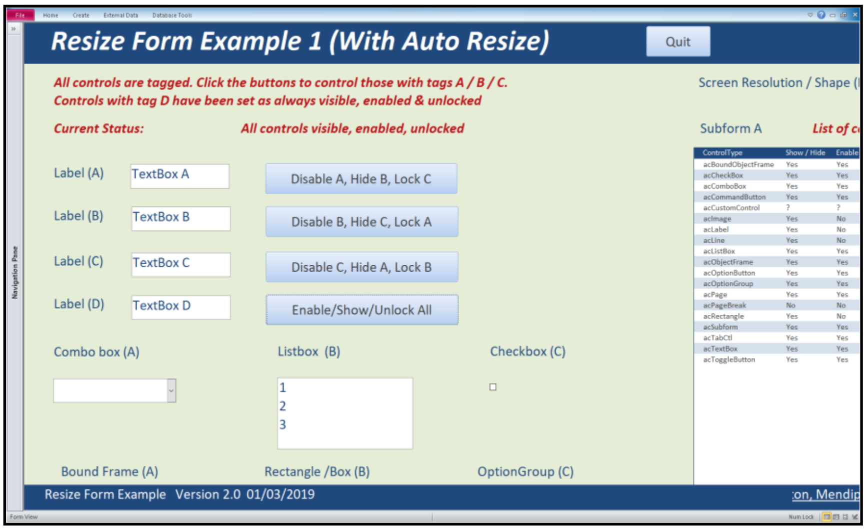
Nevertheless, it is STRONGLY recommended that forms are both designed and displayed using the same display option to help prevent any similar issues in your own applications.

---

**Datasheets** can NOT be resized. Recommend instead using a **continuous form** to emulate a datasheet

**Button captions** may not quite fit when forms are resized for small screens such as a **tablet**
To prevent this issue, ensure the **button width** is slightly greater than the length of the caption.

**Option groups** can occasionally become **'over enlarged'** switching between **form view & design view** during the development process. This can cause the entire form to expand far more than required

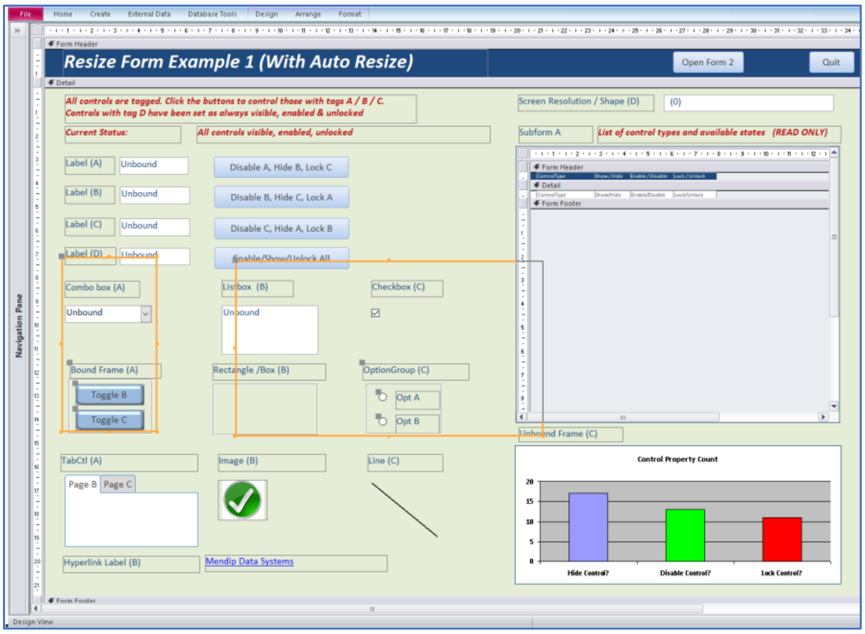


*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

*Current Status:* *All controls visible, enabled, unlocked*

| ControlType | Show / Hide | Enable |
|---|---|---|
| acBoundObjectFrame | Yes | Yes |
| acCheckBox | Yes | Yes |
| acComboBox | Yes | Yes |
| acCommandButton | Yes | Yes |
| acCustomControl | ? | ? |
| acImage | Yes | No |
| acLabel | Yes | No |
| acLine | Yes | No |
| acListBox | Yes | Yes |
| acObjectFrame | Yes | Yes |
| acOptionButton | Yes | Yes |
| acOptionGroup | Yes | Yes |
| acPage | Yes | Yes |
| acPageBreak | No | No |
| acRectangle | Yes | No |
| acSubform | Yes | Yes |
| acTabCtl | Yes | Yes |
| acTextBox | Yes | Yes |
| acToggleButton | Yes | Yes |

The cause of the problem can easily be seen in **design view**.
The **option group 'frames'** (shown in orange) have become over enlarged.
This causes the whole form to scale up more than necessary



*All controls are tagged. Click the buttons to control those with tags A / B / C.*
*Controls with tag D have been set as always visible, enabled & unlocked*

Luckily this is a rare event if the form is well designed.

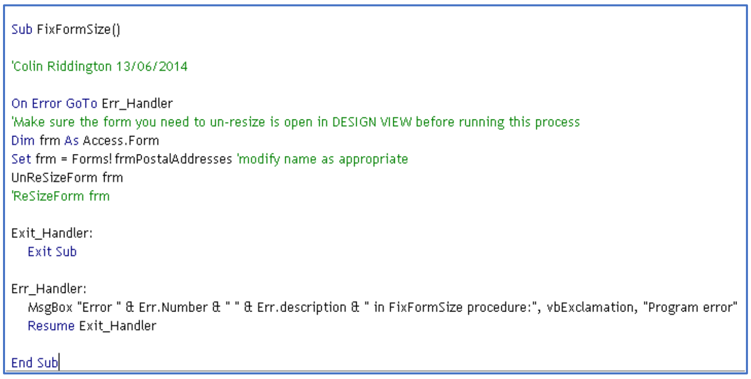
However, it can be an issue if the **option group** is near the **bottom or right edge** of the form.
To prevent this issue, **place option groups as far left and near to top of the form section** as possible.

The following quote is taken from the help file supplied with the original code by Jamie Czernik:
*Tab controls and options groups are difficult to resize as Access tries to keep the child controls within the frame while the child controls are being moved and resized. This can lead to distortion if the controls are too close to either the top/bottom or left/right of the form being resized.*

*If your forms tend to be scaled up to higher resolutions then try to keep tab controls and option groups as far left and near to top of the form section as possible. The reverse is true if forms tend to be scaled down to lower resolutions so in this situation try to keep them as far right and near to the bottom as possible.*

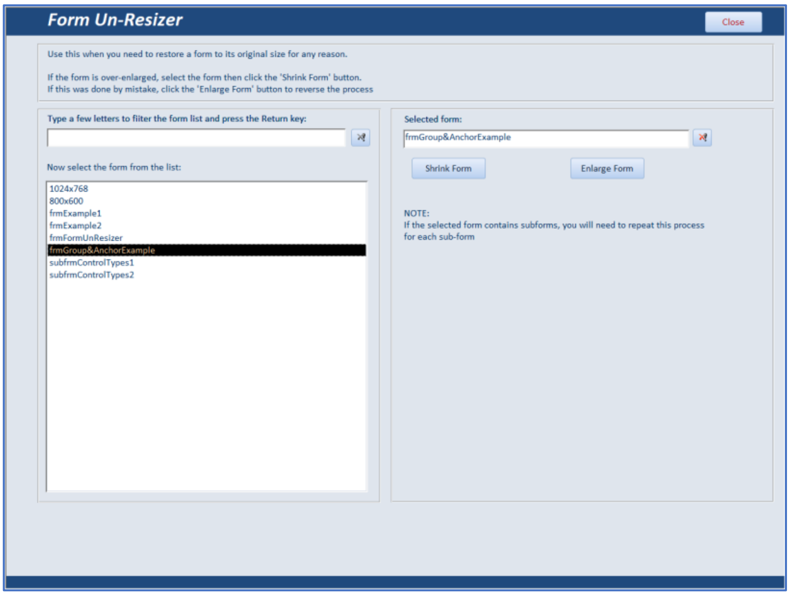
If a form does become 'over enlarged' it can be fixed using one of these methods:

a) Open the **FixFormSize** procedure and enter the form name where indicated
Run the procedure whilst the form is open in **design view**



```
Sub FixFormSize()

'Colin Riddington 13/06/2014

On Error GoTo Err_Handler
'Make sure the form you need to un-resize is open in DESIGN VIEW before running this process
Dim frm As Access.Form
Set frm = Forms!frmPostalAddresses 'modify name as appropriate
UnReSizeForm frm
'ReSizeForm frm

Exit_Handler:
   Exit Sub

Err_Handler:
   MsgBox "Error " & Err.Number & " " & Err.description & " in FixFormSize procedure:", vbExclamation, "Program error"
   Resume Exit_Handler

End Sub
```

b) Use the form **frmFormUnresizer.** Select the over enlarged form from the **list** then click **Shrink Form**



Use this when you need to restore a form to its original size for any reason.

If the form is over-enlarged, select the form then click the 'Shrink Form' button.
If this was done by mistake, click the 'Enlarge Form' button to reverse the process

Type a few letters to filter the form list and press the Return key:

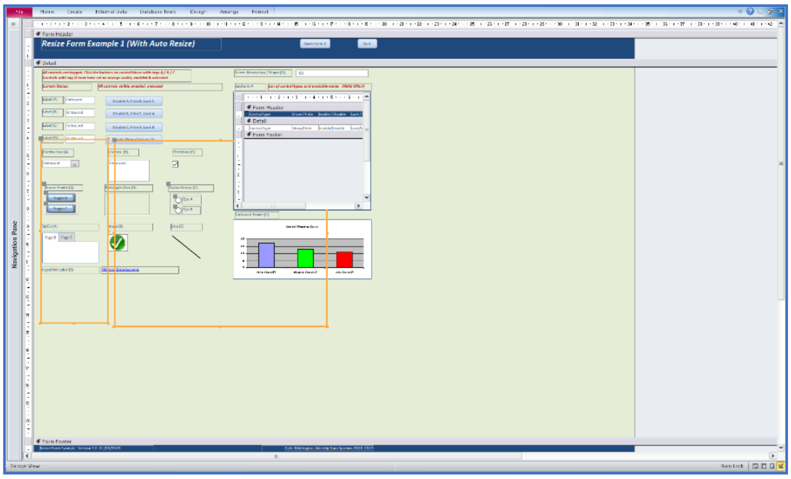
Now select the form from the list:

1024x768
800x600
frmExample1
frmExample2
frmFormUnResizer
frmGroup&AnchorExample
subfrmControlTypes1
subfrmControlTypes2

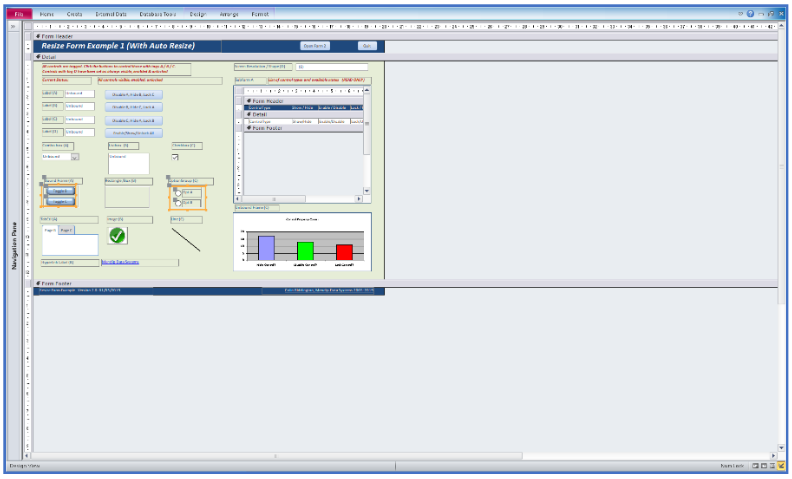
Selected form: frmGroup&AnchorExample

NOTE:
If the selected form contains subforms, you will need to repeat this process for each sub-form

In each case the form will be scaled down and controls shifted back into position.



However, you will still need to **restore the original size** of the **option group controls** that were over enlarged then **reduce the height and width** of the form yourself.



Finally **save** and **close** the form

NOTE: If necessary, the process can be reversed by clicking the **Enlarge Form** button

## ***Using form resizing with your own applications***

You will need to import the module **frmResizeForm** and (optionally) the form **frmFormUnresizer.**

Then follow the instructions as above adding the line **ResizeForm Me** to the **Form_Load** event of all forms / subforms you wish to resize.

---

**Related items**:

- The original **auto form resize** utility by **Jamie Czernik** – **afr.zip (**MDB zipped)
- The example app referenced in this article - **ResizeFormExample_v2.6.zip** (ACCDB file zipped)
- This PDF document – **ResizeForm Me_v2.6.pdf**
- The **latest version** of this document is available online at**:** http://www.mendipdatasystems.co.uk/automatic-form-resizing-1/4594554784

If you have any comments or questions, please contact me using the website feedback form or by email (see contact page on website)

*Colin Riddington* *Mendip Data Systems* *14/03/2019*