

How to reorder the specification tree with VBA

Version 2.0

Last revised: 4-14-16

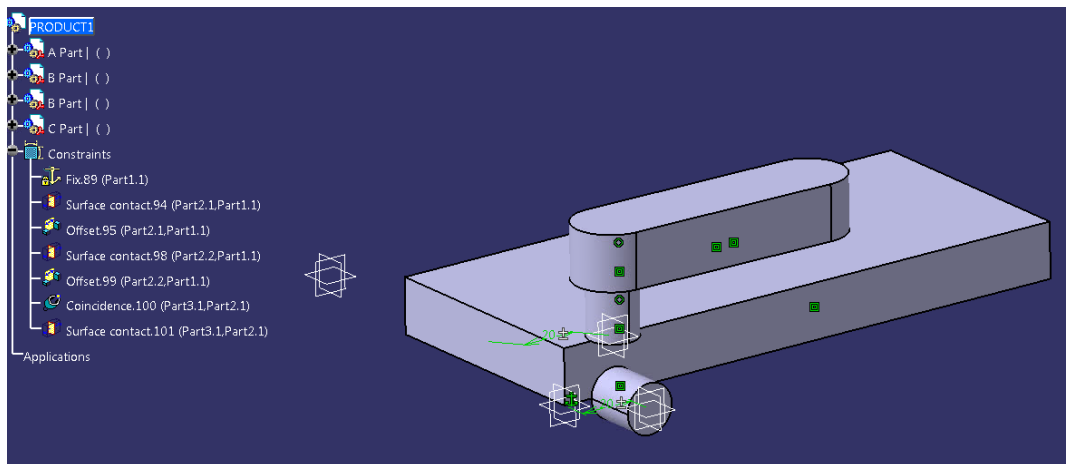
Emmett Ross

<http://www.scripting4v5.com>

This step-by-step tutorial will show you how to rearrange the order of the components in the specification tree using a VBA macro. This method will use cut and paste but the constraint links will be preserved. In this tutorial, you'll also learn how to:

- create a userform with listboxes, labels, and command buttons
- make a listbox not selectable
- keep constraints when reordering by cutting and pasting
- automatically sort the tree from A to Z or Z to A

This code reorders components under a CATProduct only, so the first step is you will either need an existing CATProduct to test this on or create a new one from scratch. For example, I made up this dummy assembly to experiment with, complete with assembly constraints:



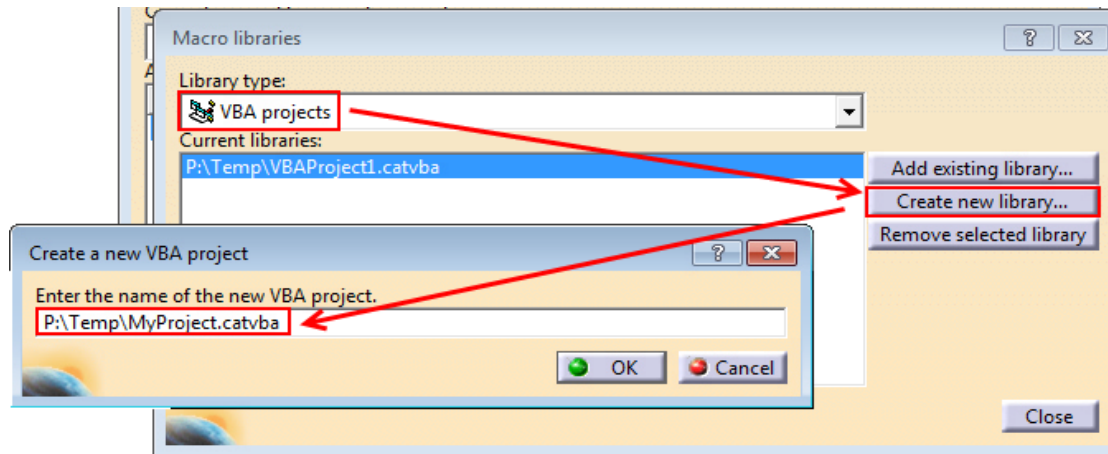
The general steps this tutorial will follow are:

1. Create a new CATProduct or open an existing assembly file in CATIA V5
2. Setup a new CATVBA program
3. Create a Userform
4. Write the code for each of the command buttons
5. Write the reorder tree code
6. Test!

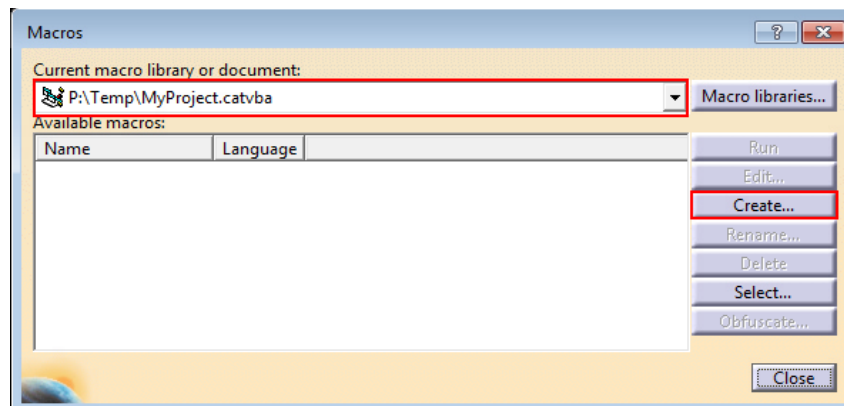
Let's get started!

Reorder Specification Tree Tutorial

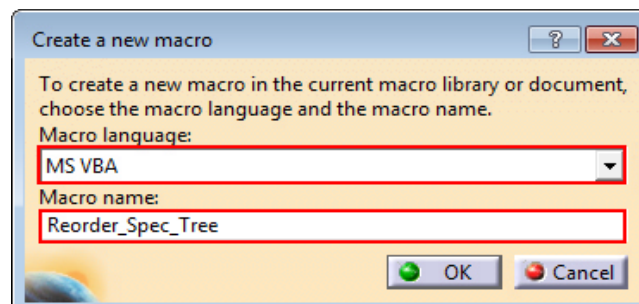
1. Go to Tools>Macro>Macros
2. Click Macro Libraries
3. Change the Library Type to VBA Projects then click Create New Library
4. Enter a file location and name for your new project, in my case: P:\Temp\MyProject.catvba



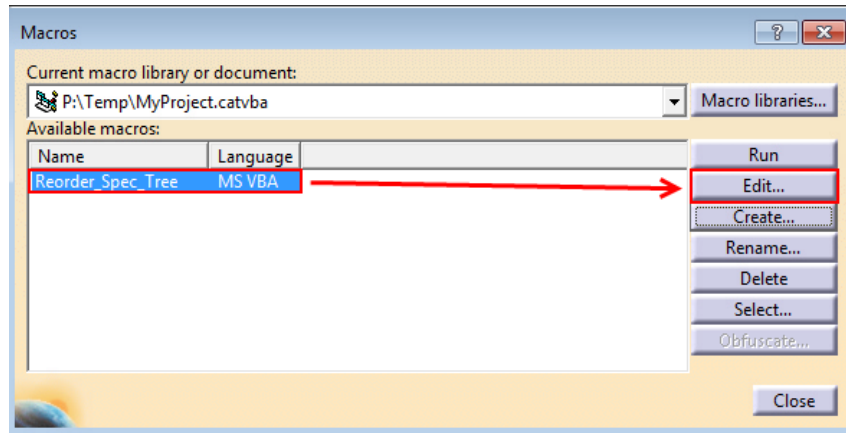
5. Click OK then close
6. Make sure your new project is the current macro library then click Create to make a new VBA macro



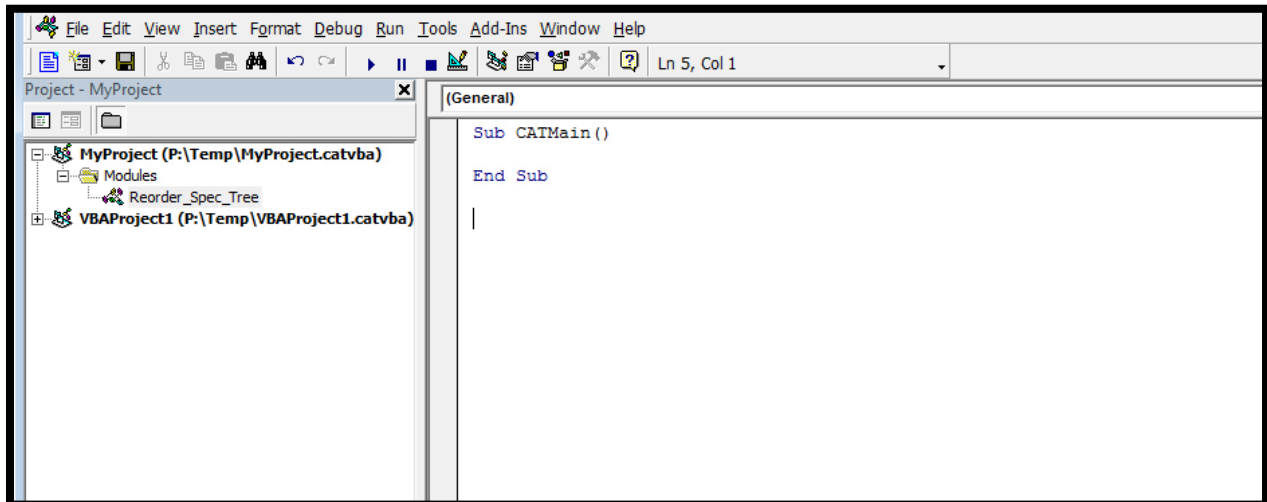
7. The macro language should be set to MS VBA. Type the macro name: Reorder_Spec_Tree



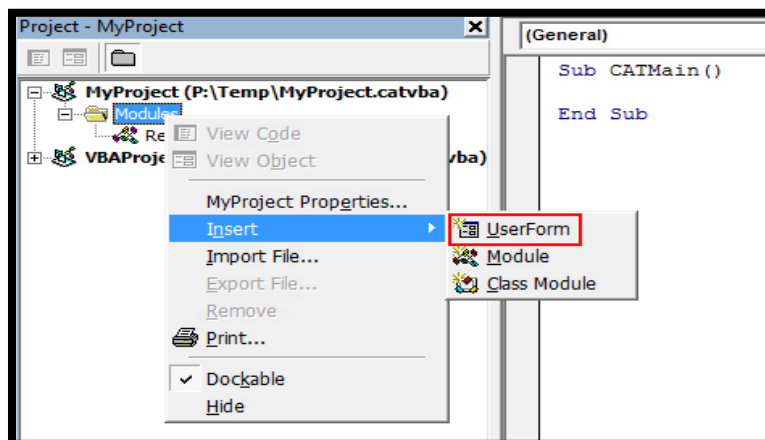
8. Select the newly created macro and click Edit to launch the VBA editor in CATIA



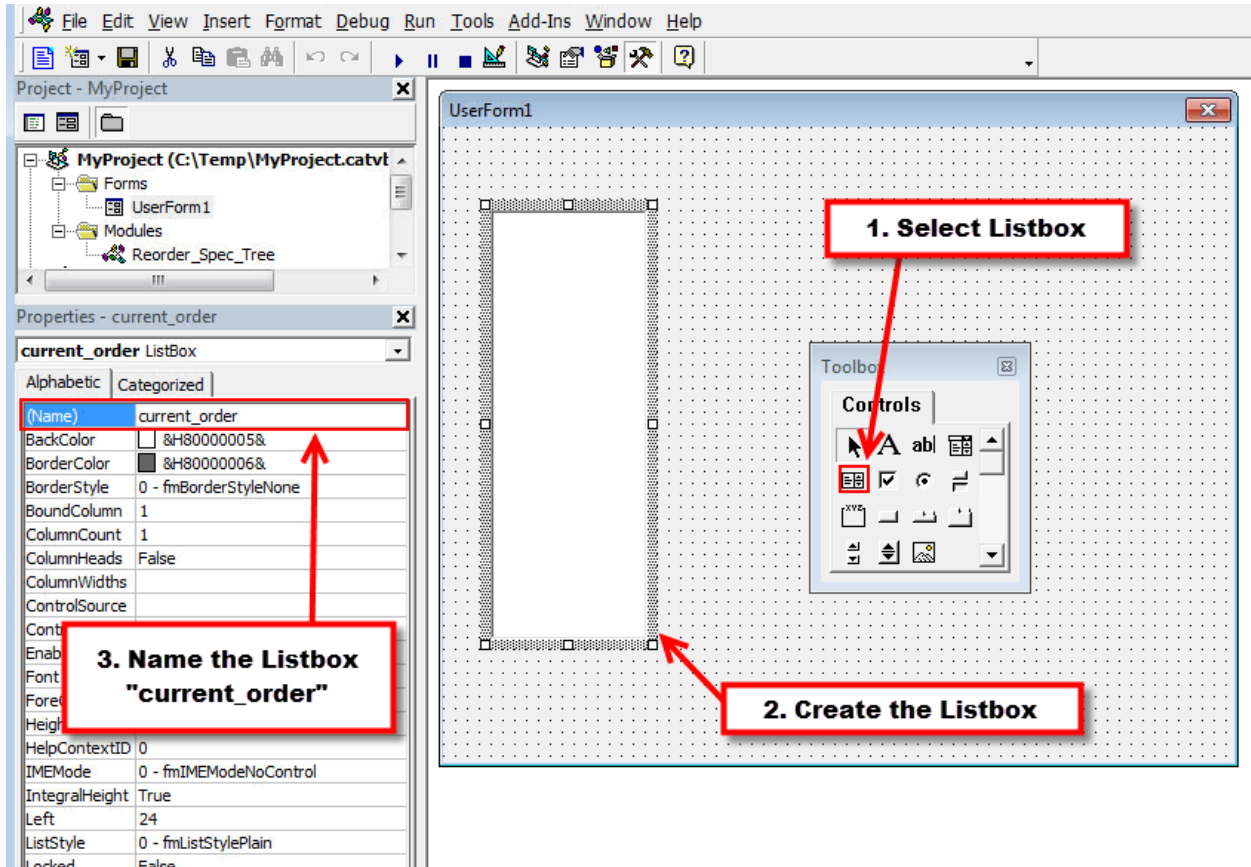
9. Your screen should now look similar to this:



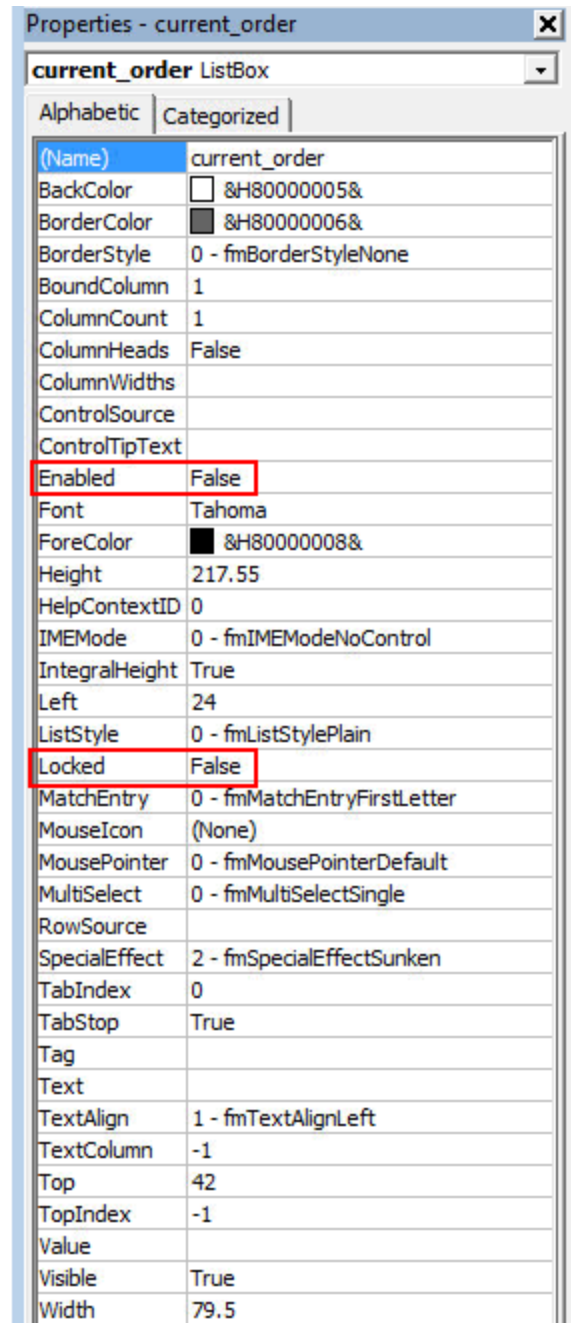
10. First, we need to create a userform that will allow the user to set the desired order of components. Right click on Module and navigate to Insert>UserForm



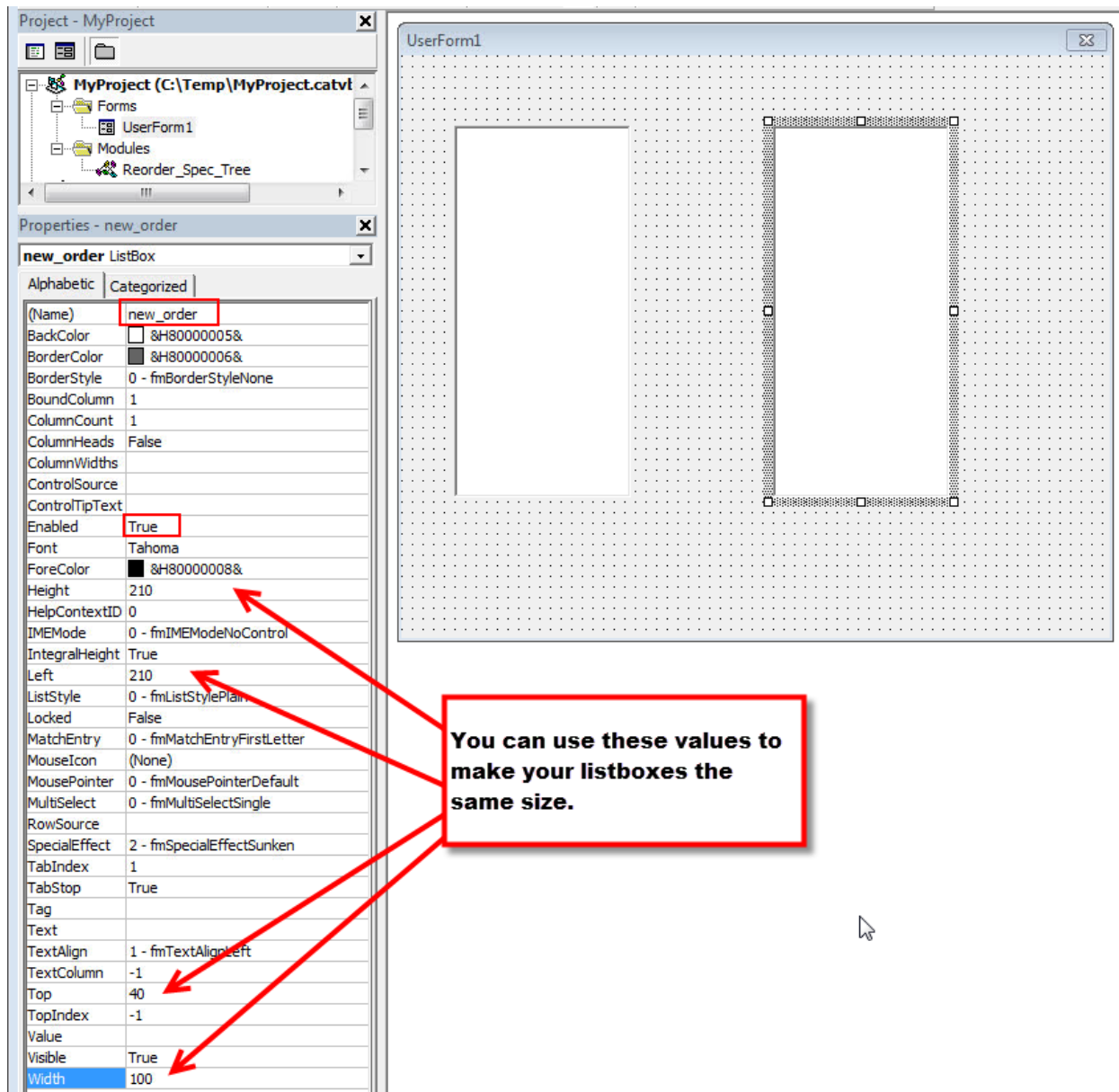
- Click on the Listbox icon within the Toolbox to create a listbox in the form. Make it any size you wish (maybe want to make it a tall box if you're dealing with a large spec tree). Ensure the name of the listbox is "current_order". This is important when we write the code for it as this listbox will display the original order of the components to the user.



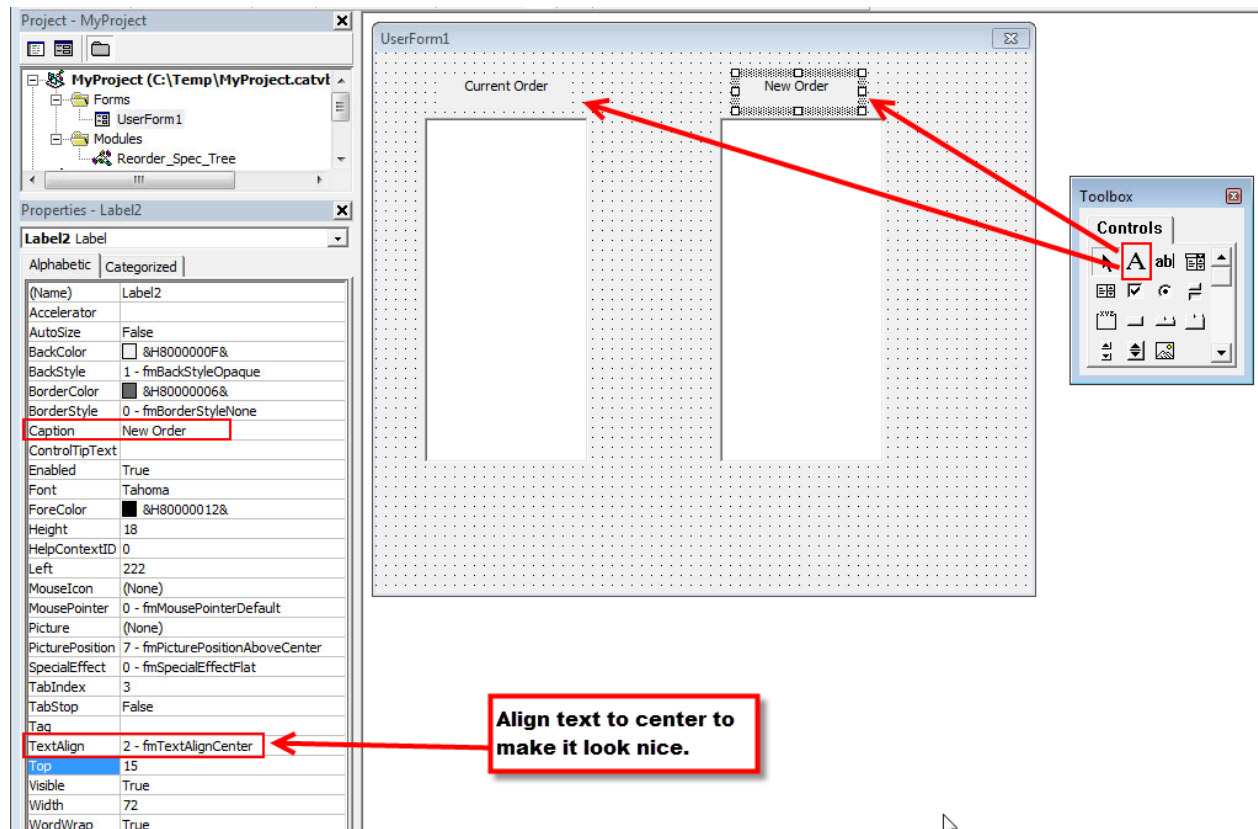
- Since this Listbox will show the original order of the components, the user does not need to select any of the values, thus we want to make the listbox non-selectable. According to <https://msdn.microsoft.com/en-us/library/office/ff869568.aspx> we need to set both the Enabled and Locked settings to False:



13. Make a second listbox, this time we will use it to sort the components into the order that we would like, so we will need to make this listbox selectable. Name it "new_order".



14. Create two labels above each listbox so the user knows which one is which (not necessary for the program to work, just a user-friendly feature).



15. Now we need to add eight (yes, 8!) command boxes to our form. Each command button needs to be labeled as shown below.

1. Reset: this button will reset the order of parts in the New Order listbox to match the Current Order listbox.
2. Up: will move a component up the tree
3. Down: will move a component down the tree
4. Sort A to Z: will automatically sort the tree from A to Z
5. Sort Z to A: will automatically sort the tree from Z to A
6. Apply: will apply the new spec tree order to the tree in CATIA
7. OK: will exit the userform with changes applied
8. Cancel: will exit the userform without making any changes in CATIA

Project - MyProject

MyProject (C:\Temp\MyProject.catv)

Forms

UserForm1

Modules

Reorder_Spec_Tree

Properties - reset_button

reset_button CommandButton

Alphabetic Categorized

(Name) reset_button

Accelerator

AutoSize False

BackColor &H8000000F&

BackStyle 1 - fmBackStyleOpaque

Cancel False

Caption Reset

ControlTipText

Default False

Enabled True

Font Tahoma

ForeColor &H80000012&

Height 24

HelpContextID 0

Left 222

Locked False

MouseIcon (None)

MousePointer 0 - fmMousePointerDefault

Picture (None)

PicturePosition 7 - fmPicturePositionAboveCenter

TabIndex 2

TabStop True

Tag

TakeFocusOnClick True

Top 258

Visible True

Width 78

WordWrap False

UserForm1

Current Order

New Order

Reset

Toolbox

Controls

Reset

Properties - up_button

up_button CommandButton

Alphabetic | Categorized

(Name)	up_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	Up
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input type="checkbox"/> &H80000012&
Height	18
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	5
TabStop	True
Tag	
TakeFocusOnClick	True
Top	60
Visible	True
Width	72
WordWrap	False

Properties - down_button

down_button CommandButton

Alphabetic | Categorized

(Name)	down_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	Down
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input type="checkbox"/> &H80000012&
Height	20
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	6
TabStop	True
Tag	
TakeFocusOnClick	True
Top	90
Visible	True
Width	72
WordWrap	False

Properties - AtoZ_button

AtoZ_button CommandButton

Alphabetic | Categorized

(Name)	AtoZ_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	Sort A to Z
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input type="checkbox"/> &H80000012&
Height	20
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	7
TabStop	True
Tag	
TakeFocusOnClick	True
Top	132
Visible	True
Width	72
WordWrap	False

Properties - ZtoA_button

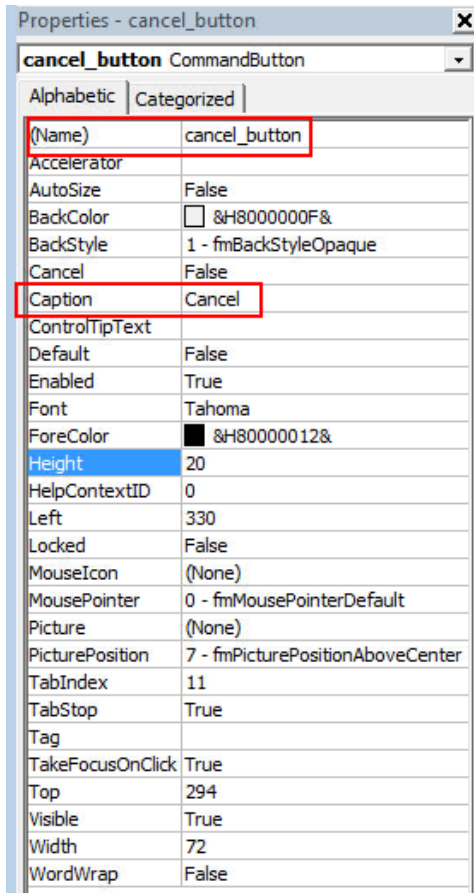
ZtoA_button CommandButton

Alphabetic | Categorized

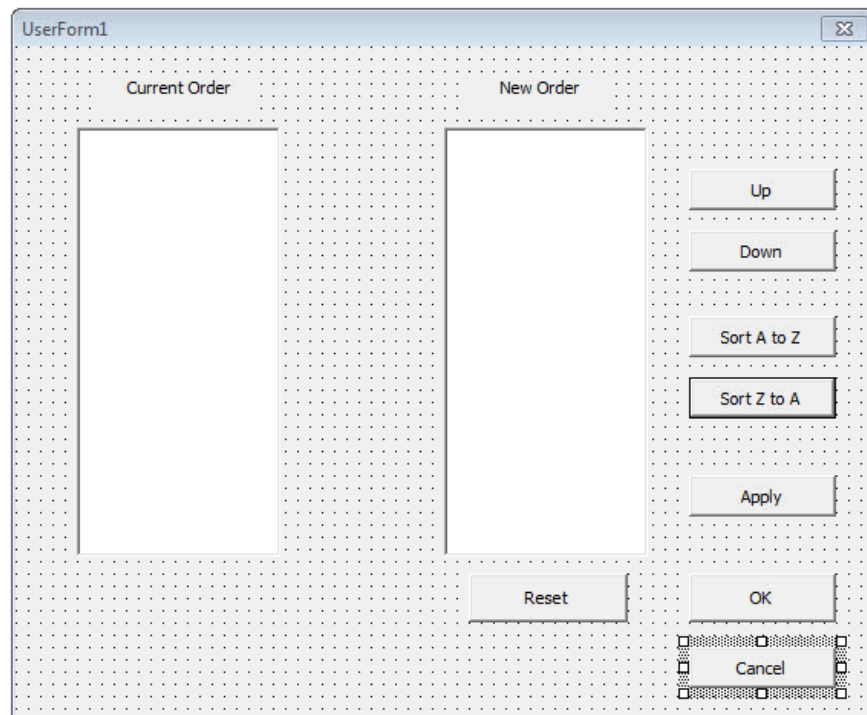
(Name)	ZtoA_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	Sort Z to A
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input type="checkbox"/> &H80000012&
Height	20
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	8
TabStop	True
Tag	
TakeFocusOnClick	True
Top	162
Visible	True
Width	72
WordWrap	False

apply_button CommandButton	
Alphabetic Categorized	
(Name)	apply_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	Apply
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input checked="" type="checkbox"/> &H80000012&
Height	20
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	9
TabStop	True
Tag	
TakeFocusOnClick	True
Top	210
Visible	True
Width	72
WordWrap	False

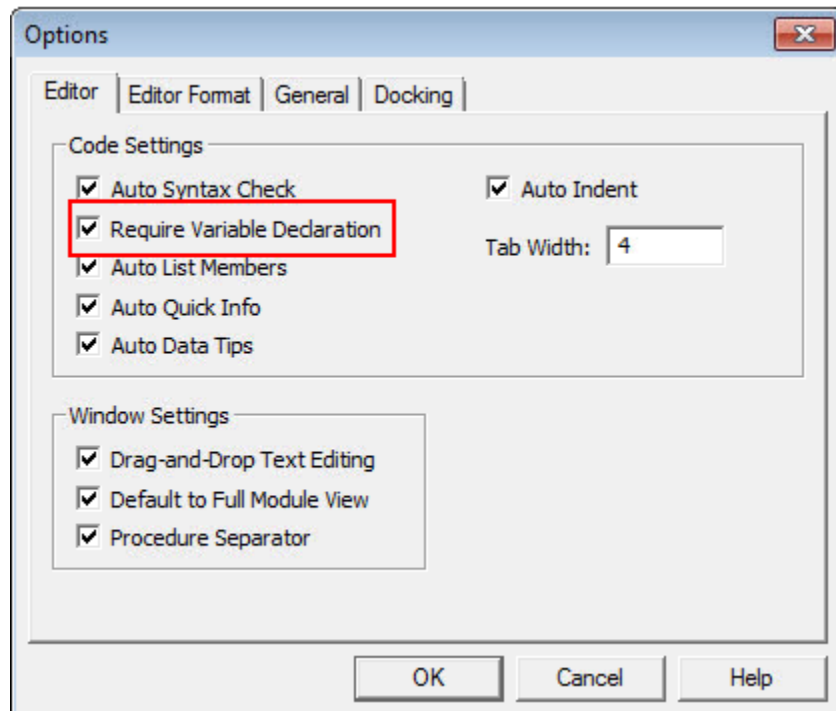
Properties - ok_button	
ok_button CommandButton	
Alphabetic Categorized	
(Name)	ok_button
Accelerator	
AutoSize	False
BackColor	<input type="checkbox"/> &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	OK
ControlTipText	
Default	False
Enabled	True
Font	Tahoma
ForeColor	<input checked="" type="checkbox"/> &H80000012&
Height	24
HelpContextID	0
Left	330
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TabIndex	10
TabStop	True
Tag	
TakeFocusOnClick	True
Top	258
Visible	True
Width	72
WordWrap	False



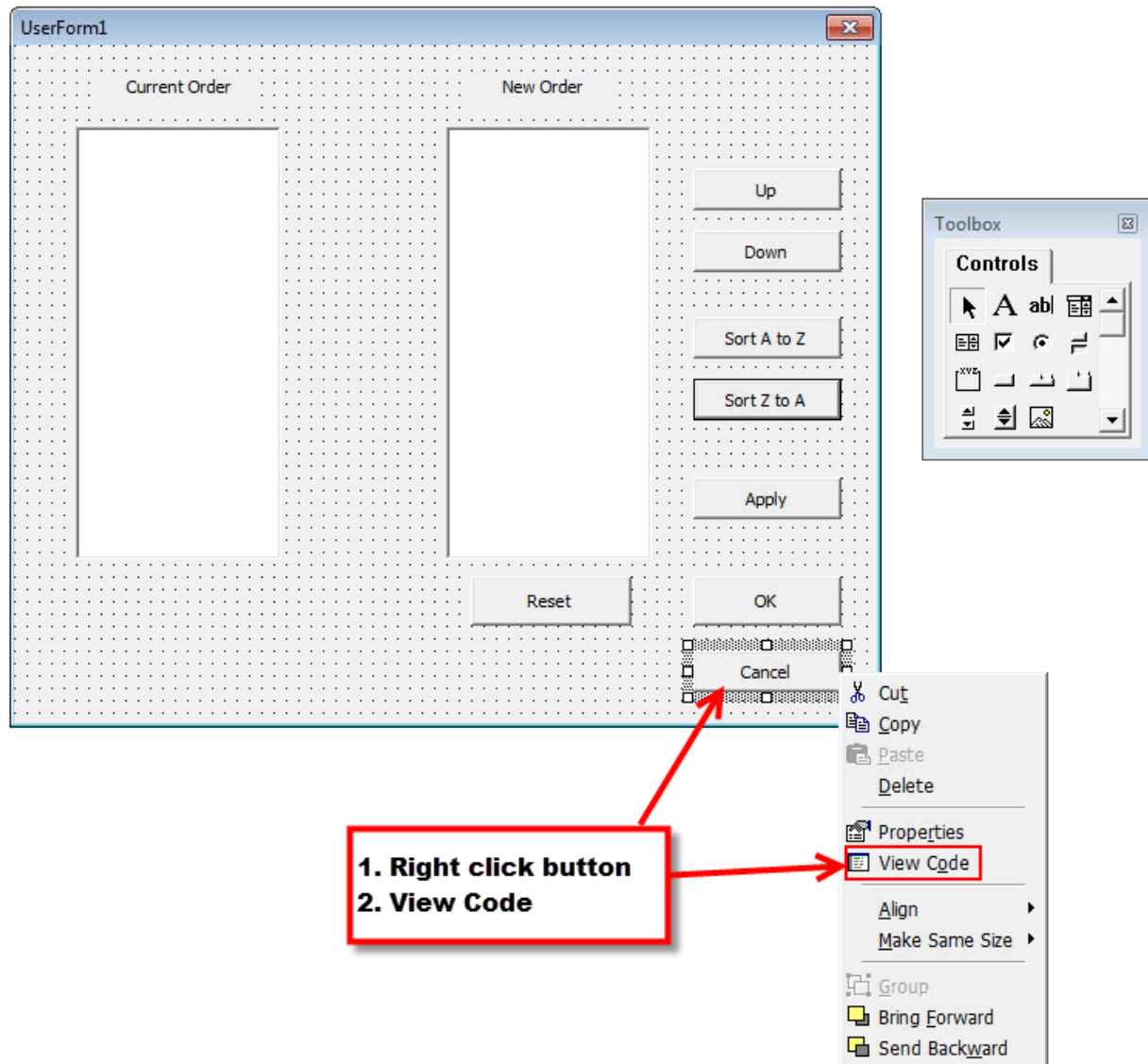
16. The completed UserForm should look like this:



17. Congratulations! You've created the userform. **Probably a good time to save your project.** Now it's time to write the code that actually makes all the command buttons work. First, let's check to make sure we are forced to declare variables by going to Tools> Options>Check "Require Variable Declaration." This will place an **Option Explicit** at the top of your code, meaning you must explicitly declare all variables by using the **Dim** or **ReDim** statements. If you try to use an undeclared variable name, an error occurs at compile time.



18. To write the code for the Cancel button, right click on it and select View Code. The code for all the userform's buttons will be contained within the same window.



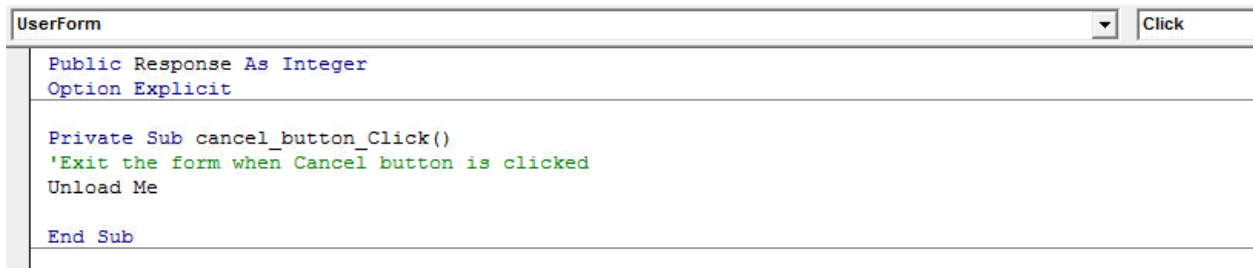
```
cancel_button Click
Private Sub cancel_button_Click()
End Sub

Private Sub UserForm_Click()
End Sub
```

Code for cancel button written here.

19. When the cancel button is clicked, it should stop the program from running and exit the form without making any changes in CATIA. Use this code:

```
Public Response As Integer 'will be used for error handling later
Option Explicit
Private Sub cancel_button_Click()
'Exit the form when Cancel button is clicked
Unload Me
End Sub
```



Check: Double click UserForm1 in the tree then press the play button to run the macro. Try pressing all the command buttons. None of them should do anything until you get to the Cancel button, which should exit the form and take you back to the VBA editor.

20. Go back to the form, right click on the OK button and select View Code.

```
Private Sub ok_button_Click()
```

```
'Error handling - if no items in the list then give error message
```

```
If new_order.ListCount = 0 Then
```

```
Response = MsgBox("There are no items in the list to reorder! Please select at least one component.", ,  
UserForm1.Name & "Warning Message")
```

```
Exit Sub
```

```
End If
```

```
'Reorders the tree and then exits the userform
```

```
Reorder_Tree
```

```
Unload Me
```

```
End Sub
```

The Reorder_Tree is going to reference another part of the code where the magic is going to happen, but does nothing for now.

```
Private Sub ok_button_Click()
```

```
'Error handling - if no items in the list then give error message
```

```
If new_order.ListCount = 0 Then
```

```
Response = MsgBox("There are no items in the list to reorder! Please select at least one component.", , UserForm1.Name & "Warning Message")
```

```
Exit Sub
```

```
End If
```

```
'Reorders the tree and then exits the userform
```

```
Reorder_Tree
```

```
Unload Me
```

```
End Sub
```


21. Next, right click on the Apply button and view code. This button will reorder the tree but will NOT exit the form and the program will stay running.

```
Private Sub apply_button_Click()  
'reorders the tree and stays in the userform  
Reorder_Tree  
End Sub
```

At this point, your code should look like this (do you see why we gave each button a unique name!):

UserForm	Click
<pre>Public Response As Integer Option Explicit Private Sub apply_button_Click() <i>'reorders the tree and stays in the userform</i> Reorder_Tree End Sub Private Sub cancel_button_Click() <i>'Exit the form when Cancel button is Clicked</i> Unload Me End Sub Private Sub ok_button_Click() <i>'Error handling - if no items in the list then give error message</i> If new_order.ListCount = 0 Then Response = MsgBox("There are no items in the list to reorder! Please select at least one component.", , UserForm1.Name & "Warning Message") Exit Sub End If <i>'Reorders the tree and then exits the userform</i> Reorder_Tree Unload Me End Sub Private Sub UserForm_Click() End Sub</pre>	

22. Moving up the buttons, we go on to the Sort Z to A button. Type this code:

```
Private Sub ZtoA_button_Click()  
Dim i As Long  
Dim j As Long  
Dim temp As Variant  
  
'sort the spec tree order from Z to A  
For j = 0 To new_order.ListCount - 2  
    For i = 0 To new_order.ListCount - 2  
        If new_order.List(i) < new_order.List(i + 1) Then  
            temp = new_order.List(i)  
            new_order.List(i) = new_order.List(i + 1)  
            new_order.List(i + 1) = temp  
        End If  
    Next i  
Next j  
End Sub
```

```
Private Sub ZtoA_button_Click()  
Dim i As Long  
Dim j As Long  
Dim temp As Variant  
  
'sort the spec tree order from Z to A  
For j = 0 To new_order.ListCount - 2  
    For i = 0 To new_order.ListCount - 2  
        If new_order.List(i) < new_order.List(i + 1) Then  
            temp = new_order.List(i)  
            new_order.List(i) = new_order.List(i + 1)  
            new_order.List(i + 1) = temp  
        End If  
    Next i  
Next j  
End Sub
```

23. To sort from A to Z (rather than Z to A), copy and paste the code is just wrote but change the direction of "<" to ">".

```
Private Sub AtoZ_button_Click()
```

```
Dim i As Long
```

```
Dim j As Long
```

```
Dim temp As Variant
```

```
'sort the spec tree order from A to Z
```

```
For j = 0 To new_order.ListCount - 2
```

```
For i = 0 To new_order.ListCount - 2
```

```
    If new_order.List(i) > new_order.List(i + 1) Then
```

```
        temp = new_order.List(i)
```

```
        new_order.List(i) = new_order.List(i + 1)
```

```
        new_order.List(i + 1) = temp
```

```
    End If
```

```
Next i
```

```
Next j
```

```
End Sub
```

```
Private Sub AtoZ_button_Click()  
Dim i As Long  
Dim j As Long  
Dim temp As Variant  
  
'sort the spec tree order from A to Z  
For j = 0 To new_order.ListCount - 2  
    For i = 0 To new_order.ListCount - 2  
        If new_order.List(i) > new_order.List(i + 1) Then  
            temp = new_order.List(i)  
            new_order.List(i) = new_order.List(i + 1)  
            new_order.List(i + 1) = temp  
        End If  
    Next i  
Next j  
End Sub
```

24. Three more buttons to go! Next, we'll add the code for the Down button. I'll use comments to explain what is happening.

```
Private Sub down_button_Click()  
Dim selector As String           'The selected item will be saved here  
Dim selected_position As Integer 'The position of the selected item will be saved here  
Dim current_occupant As String   'The item one step underneath the selected item will be saved here  
  
'----- variable initialization -----  
selector = ""  
selected_position = 0  
  
current_occupant = ""  
  
'selected item and its position are determined  
selected_position = new_order.ListIndex  
selector = new_order.List(selected_position)  
  
'current_occupant is determined  
'(IF-statement makes sure nothing happens if item in the very bottom is selected)  
If selected_position < (new_order.ListCount - 1) Then  
    current_occupant = new_order.List(selected_position + 1)  
  
    'move the selected item down by one position  
    new_order.List(selected_position + 1) = selector  
    new_order.List(selected_position) = current_occupant  
  
    'Keep the selected item selected (otherwise the current_occupant,  
    'that has switched places with selector, will be selected)  
    new_order.Selected(selected_position + 1) = True  
End If  
End Sub
```

25. The code for the Up button is very similar to the Down button, it just moves the components in the tree the opposite direction.

```
Private Sub up_button_Click()  
    Dim selector As String          'The selected item will be saved here  
    Dim selected_position As Integer 'The position of the selected item will be saved here  
    Dim current_occupant As String  'The item one step above the selected item will be saved here  
  
    selector = ""  
    selected_position = 0  
    current_occupant = ""  
  
    'selected item and its position are determined  
    selected_position = new_order.ListIndex  
    selector = new_order.List(selected_position)  
  
    'current_occupant is determined  
    '(IF-statement makes sure nothing happens if top-most item is selected)  
    If selected_position > 0 Then  
        current_occupant = new_order.List(selected_position - 1)  
  
        'move the selected item up by one position  
        new_order.List(selected_position - 1) = selector  
        new_order.List(selected_position) = current_occupant  
  
        'Keep the selected item selected (otherwise the current_occupant,  
        'that has switched places with selector, will be selected)  
        new_order.Selected(selected_position - 1) = True  
    End If  
End Sub
```

26. Finally! The last button is the Reset button to reset the new order to be the same as the current order.

```
Private Sub reset_button_Click()
```

```
'this button resets the new_order to the current_order
```

```
Dim i As Integer
```

```
new_order.Clear
```

```
i = 0
```

```
For i = 0 To current_order.ListCount - 1
```

```
    new_order.AddItem current_order.List(i)
```

```
Next i
```

```
End Sub
```

```
Private Sub reset_button_Click()  
'this button resets the new_order to the current_order  
  
Dim i As Integer  
  
new_order.Clear  
  
i = 0  
  
For i = 0 To current_order.ListCount - 1  
    new_order.AddItem current_order.List(i)  
Next i  
End Sub
```

27. That covers all eight of our command buttons. Whew! If you remember, we had a line of code in some of the commands, Reorder_Tree. It's time to add the code for which this is referring. This is where the reordering magic happens. Add this code below all the command button code:

```
Sub Reorder_Tree()  
  
Dim selection1 As Selection  
Dim selection2 As Selection  
Dim ProdName As String  
Dim j As Integer  
Dim i As Integer  
  
'---keep assembly constraints when cutting and pasting-----  
Dim settingControllers1 As SettingControllers  
Set settingControllers1 = CATIA.SettingControllers  
Dim asmConstraintSettingAtt1 As AsmConstraintSettingAtt  
Set asmConstraintSettingAtt1 = settingControllers1.Item("CATAsmConstraintSettingCtrl")  
asmConstraintSettingAtt1.PasteComponentMode = catPasteWithCstOnCopyAndCut  
  
'-----  
Set selection1 = productDocument1.Selection  
  
'This loop goes through the items in new_order  
'and compares them to the components of the selected product.  
For j = 0 To UserForm1.new_order.ListCount - 1  
  
    'Getting selection1 ready for the operation by clearing whatever might be in it  
    selection1.Clear  
  
    'This for loop goes through the items of the selected product  
    For i = 1 To product1.Products.Count  
  
        ProdName = product1.Products.Item(i).Name  
  
        If Left(ProdName, Len(new_order.List(j))) = new_order.List(j) Then  
            selection1.Add product1.Products.Item(i)  
        End If  
  
    Next i  
  
Next i
```

'cut and paste the components from the original order to the new order

selection1.Cut

Set selection2 = CATIA.ActiveDocument.Selection

selection2.Add product1

selection2.Paste

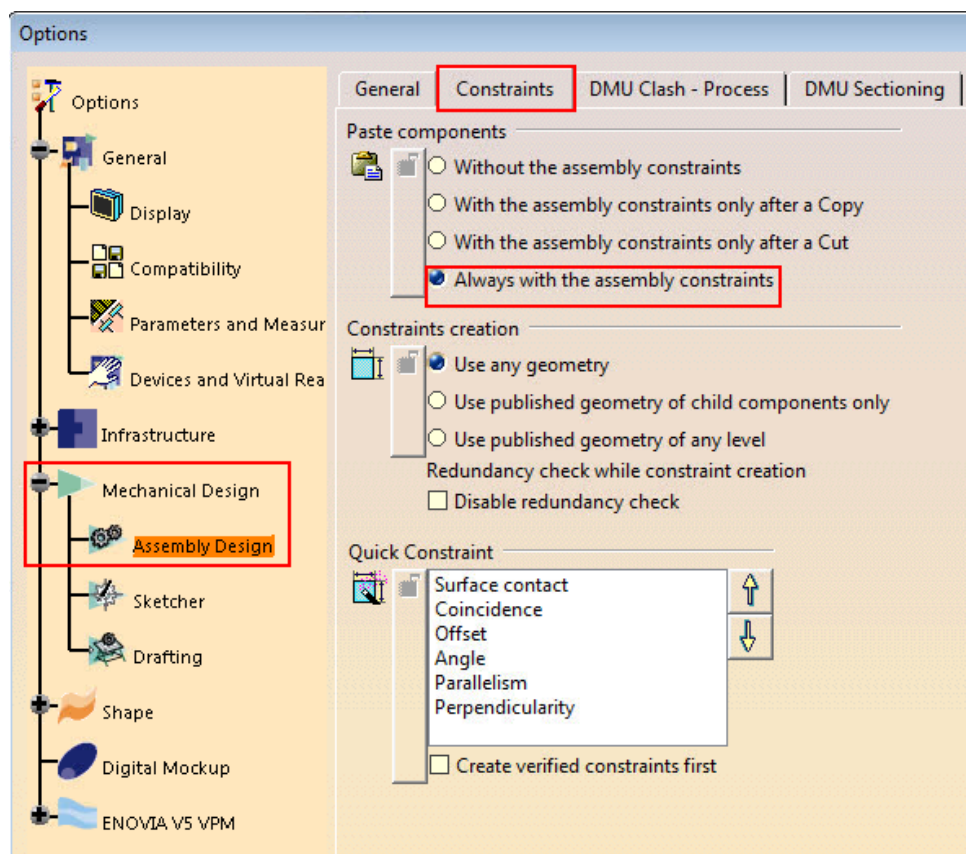
selection1.Clear

selection2.Clear

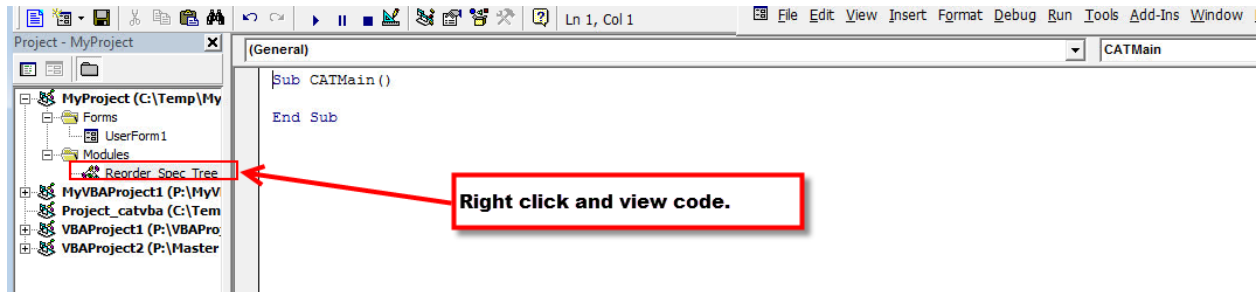
Next j

End Sub

You may have noticed the section on keeping the constraints when cutting and pasting. This is an option in CATIA V5 under Options>Mechanical Design>Assembly Design> tab: Constraints>Paste components>Always with the assembly constraints. We added code to programmatically select this option every time so the assembly constraints will not get broken.



28. So now all the buttons should be functioning properly, the problem is we need to populate the first list box with the names of all the components. To do this, we're going to go back to our original and still blank module Reorder_Spec_Tree (again, you can right click and select View Code).



Add the code shown below to the CATMain then save it. This is where the macro will be run from.

Option Explicit

```
Public productDocument1 As Document      'Holds the active document
Public product1 As Product               'Holds the main product
Public products1 As Products             'Holds the list of component parts of product1
Public selection1 As Selection           'For looping operations
Public check As Boolean                  'For looping operations
Public ip, jp, lp, Response1 As Integer 'for looping operations
```

```
Sub CATMain()
Dim str As String                       'Used to save name of element for listing in the ListBox
```

```
Set productDocument1 = CATIA.ActiveDocument 'productDocument1 set to hold current document
'products1 set to hold the main product of the active document
Set product1 = productDocument1.Product
Set products1 = product1.Products          'products1 set to hold the list of components of product1
```

```
'loop through all components to add them to the listbox
```

```
For ip = 1 To products1.Count
    check = False
    lp = InStr(products1.Item(ip).Name, ".")
    str = Left(products1.Item(ip).Name, lp - 1)
```

```
'add all component names to first listbox
```

```
For jp = 0 To UserForm1.current_order.ListCount - 1
```

```
If str = UserForm1.current_order.List(jp) Then  
    check = True  
End If  
Next jp
```

```
If check = False Then  
    UserForm1.current_order.AddItem str  
End If
```

```
Next ip
```

```
'Finally new_order is also populated  
UserForm1.new_order.Clear
```

```
ip = 0
```

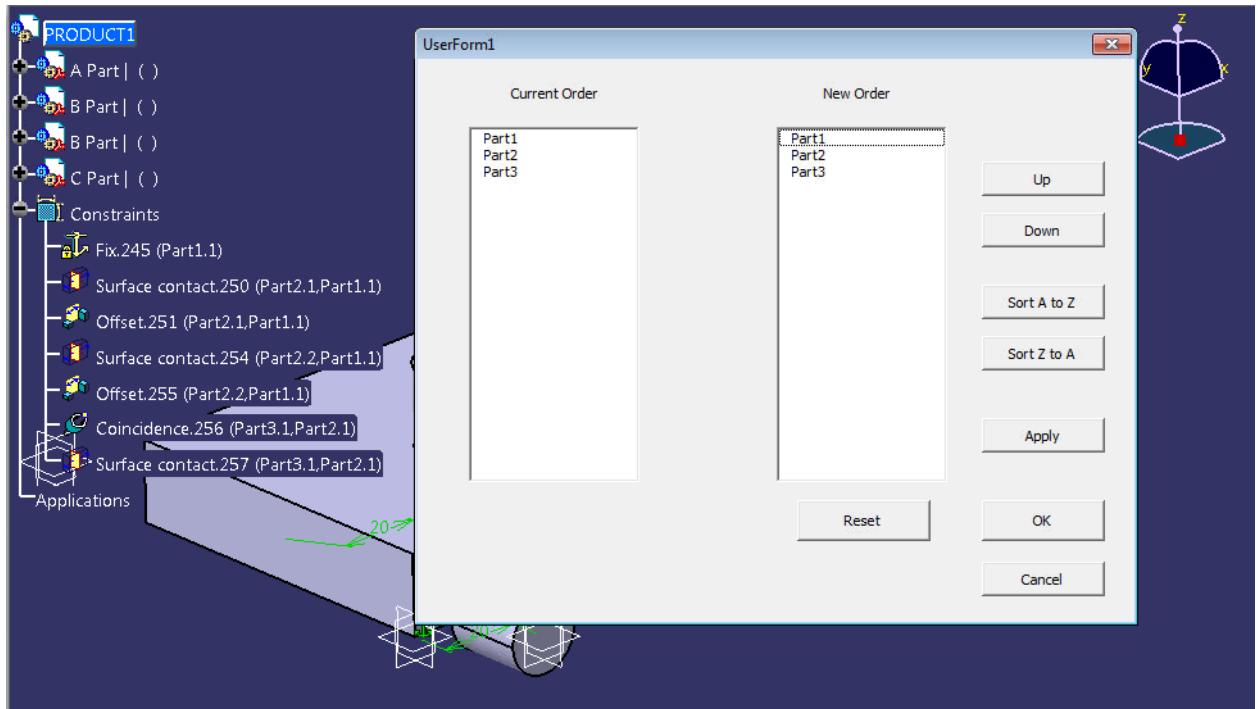
```
For ip = 0 To UserForm1.current_order.ListCount - 1  
    UserForm1.new_order.AddItem UserForm1.current_order.List(ip)  
Next ip
```

```
UserForm1.Show
```

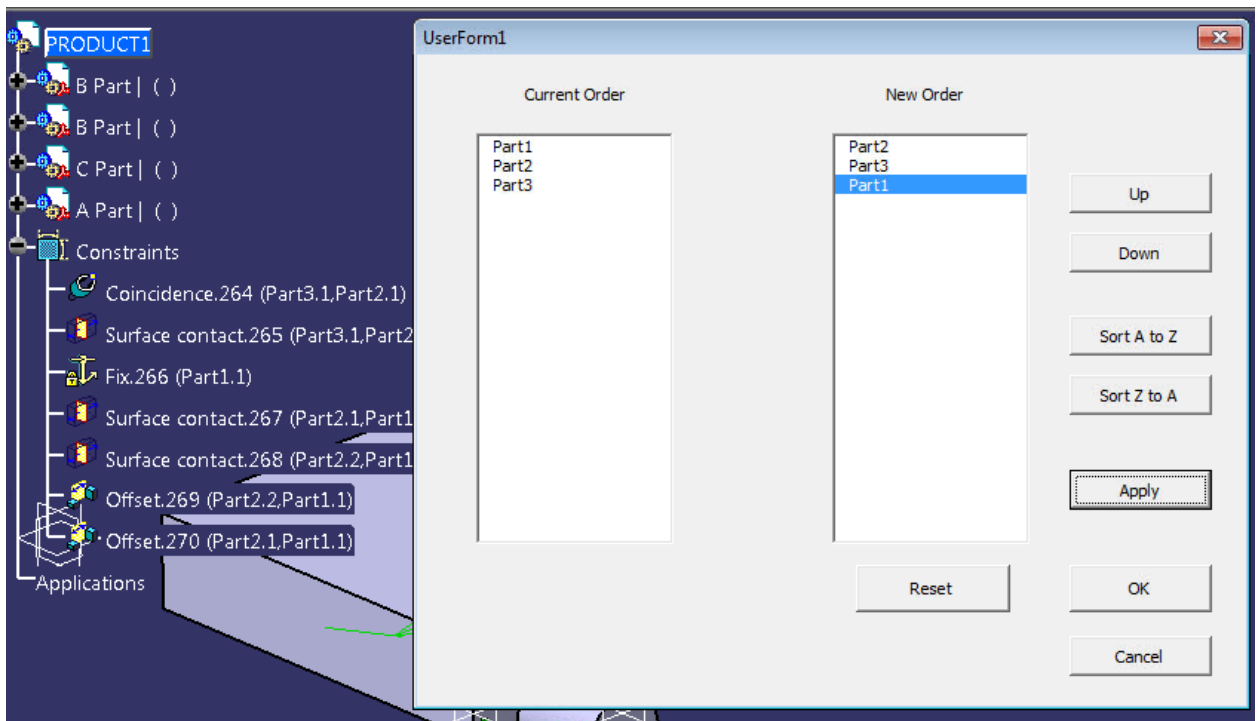
```
End Sub
```

29. That's it! Time to run the Reorder Spec Tree macro on your CATProduct file. You should see the list of parts appear in the first Listbox.

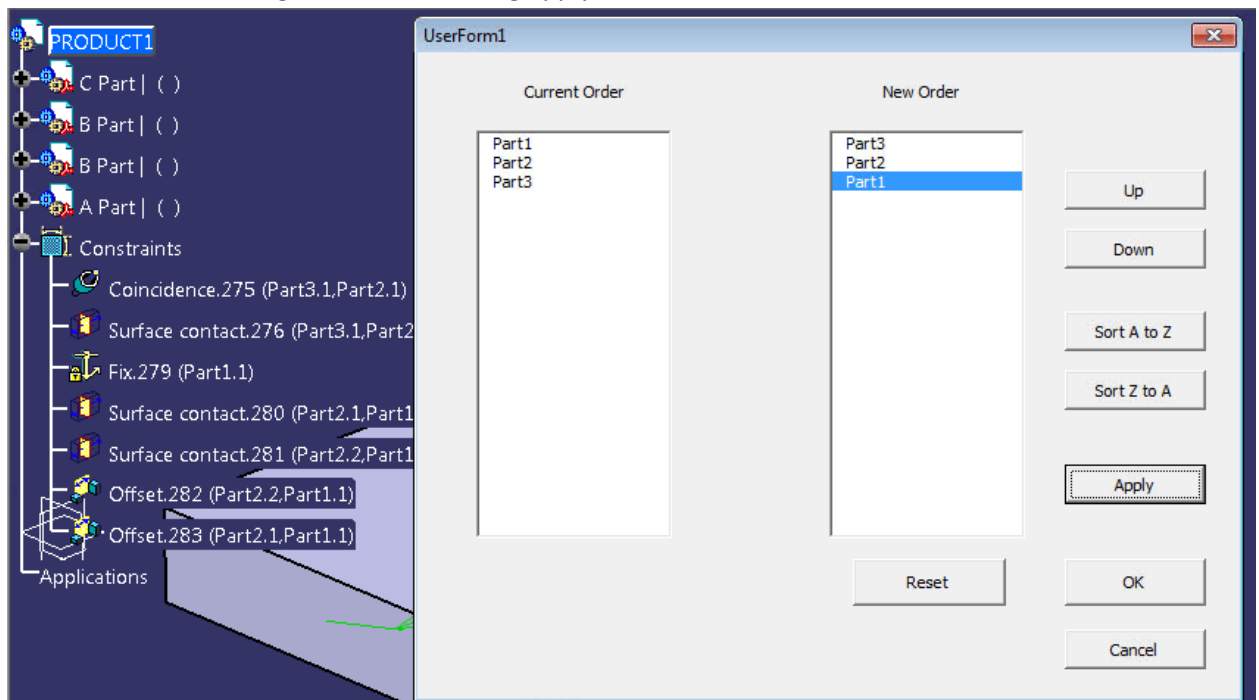
Screenshot is just after launching the program:



Screenshot after manually click the up and down buttons followed by Apply:



Screenshot after sorting Z to A then clicking apply:



Video Demonstration

To see a video of this program in action, check this out:

<https://youtu.be/1ayn0DmINR0>

###

I hope you enjoyed this tutorial as much as I enjoyed making it for you!

Questions or comments? Email me: Emmett@scripting4v5.com

For more articles, tutorials, and downloads, please visit <http://www.scripting4v5.com>

I worked hard on this project, please don't share with anyone who hasn't purchased it. In fact, if you want to refer your friends and coworkers I will give you a commission. Send me an email if you're interested in becoming an affiliate.