





















VBA SOURCE CODE BOOK

PHYSICAL INVENTORY WORKSHEET - **UPDATED**

PHYSICAL INVENTORY WORKSHEET

Filter By: Item Cost Sort By: Item Name Total Adjusted Value: \$ -

Item Name	Part #	Category	Sales Description	Item Cost	Qty In Stock	Bar Code	Picture	Actual Qty	Value Adj.	Item
Bath Vanity Set	1D523	Assembly	Bathroom Vanity Set	\$1,899.94	1					
Bathroom Vanity 50" 1D515	1D515	Cabinets	Assemble & Install Bathroom Vanity Cabir	\$299.00	2					
ADJUSTMENT HISTORY										
Adj. Date	Actual Qty	Adj. Qty	umbering	Cast Iron Bathtub - White	\$399.00	5				
2/7/2022	4	2								
2/7/2022	1	-1	pplies	Apply Silicone Caulking	\$3.99	3				
2/7/2022	4	2								
2/7/2022	2	0	omputer	Computer Fan 120 Volt *	\$25.00	18				
Counter White 55"	1D517	Cabinets	Install Vanity Marble Countertop - 55"	\$295.00	8					
Faucet SinHa Chrm	1D518	Plumbing	Install 2 Handle Chrome Faucet	\$95.00	10					
Full Computer Set	F5458	Assembly	Lenovo Full Computer Assembly	\$528.99	3					
Graphics Card 16G	F4517	Computer	Graphics Card 16GB DD	\$78.00	18					
Hard Drive 1TB SSD	F4518	Computer	Hard Drive 1TB SSD Compact	\$68.00	18					

EXCLUSIVE FOR SILVER & GOLD MEMBERS

FEATURE, FIX, OR FOCUS



Excel For Freelancers

How To Create A Physical Inventory Worksheet With Barcodes & Pictures In Excel From Scratch



[DOWNLOAD
APPLICATION](#)



[VIEW
TRAINING](#)

by: Randy Austin

ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With nearly 300,000 YouTube subscribers, 14,000,000 video views, 200+ comprehensive training videos, and a thriving 40,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®
Most Valuable
Professional



OUR COURSES & PRODUCTS



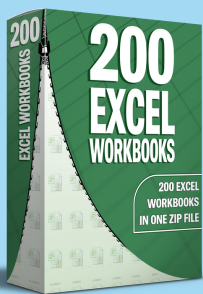
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



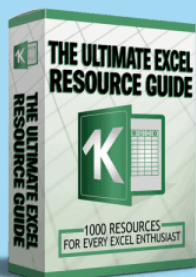
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



Incredible Package of 200 of my BEST Applications into a SINGLE ZIP File which also includes the "200 Workbook Library".

[Click here to learn more](#)



With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)

Projects.....2
VBAProject.....2
Documents.....2
 AdjItems.....2
 (Declarations).....2
Adjust.....3
 (Declarations).....3
InvWkSht.....4
 (Declarations).....4
 Worksheet_Change [Sub].....4
 Worksheet_SelectionChange [Sub].....4
Items.....5
 (Declarations).....5
ThisWorkbook.....6
 (Declarations).....6
Modules.....7
 InvAdj_Macros.....7
 (Declarations).....7
 AdjustItemHistory [Sub].....7
 InvAdj_ClearFilter [Sub].....7
 InventoryAdj_AddPictures [Sub].....7
 InventoryAdj_Load [Sub].....7
 InventoryAdj_LoadItems [Sub].....8
 InventoryAdj_New [Sub].....9
 InventoryAdj_Save [Sub].....9
 NextAdj [Sub].....10
 PrevAdj [Sub].....10
 Print_InventoryWkSht [Sub].....10
 ShowAdjHistory [Sub].....11

1 Option Explicit

2

1 Option Explicit

2

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("L7:L9999" )) Is Nothing And Target.CountLarge < 2
5         Then Range("B2" ).Value = True
6         If Not Intersect(Target, Range("F4" )) Is Nothing And Range("F4" ).Value <> Empty
7             Then InventoryAdj_LoadItems
8         If Not Intersect(Target, Range("I4" )) Is Nothing And Range("I4" ).Value <> Empty
9             Then InventoryAdj_LoadItems
10        End Sub
11
12 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
13     If Shapes("ItemHistBtn" ).Visible = True Then Shapes("ItemHistBtn" ).Visible =
14         msoFalse 'Hide Item If visible
15     If Shapes("AdjusHistoryPopUp" ).Visible = True Then Shapes("AdjusHistoryPopUp" ).
16         Visible = msoFalse 'Hide picture if visible
17     If Not Intersect(Target, Range("D7:N9999" )) Is Nothing And Range("D" & Target.Row)
18         .Value <> Empty Then
19         Range("P5" ).Value = Target.Row
20         AdjItems.Range("V3" ).Value = Range("A" & Target.Row) 'Item ID Criteria
21         With Shapes("ItemHistBtn" )
22             .Left = Range("C" & Target.Row).Left
23             .Top = Range("C" & Target.Row).Top
24             .Visible = msoCTrue
25         End With
26         AdjustItemHistory
27     End If
28 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

```

1 Option Explicit
2 Dim PicFolder As String, PicPath As String
3 Dim LastRow As Long, ItemRow As Long, LastResultRow As Long, AdjDBRow As Long, AdjRow
  As Long, ItemDBRow As Long, AdjID As Long, SortCol As Long
4 Dim PicShp As Shape
5
6 Sub AdjustItemHistory()
7     With AdjItems
8         LastRow = .Range("A99999").End(xlUp).Row
9         If LastRow < 3 Then Exit Sub
10        .Range("A2:G" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
11        "V2:V3"), CopyToRange:=.Range("W2:Z2"), Unique:=False
12        LastResultRow = .Range("W99999").End(xlUp).Row
13        If LastResultRow < 3 Then Exit Sub
14    End With
15 End Sub
16
17 Sub InvAdj_ClearFilter()
18     InvWkSht.Range("F4:G4").ClearContents
19     InventoryAdj_LoadItems
20 End Sub
21
22 Sub InventoryAdj_AddPictures()
23     With InvWkSht
24         LastRow = .Range("A99999").End(xlUp).Row
25         PicFolder = .Range("P4").Value & "\"
26         For ItemRow = 7 To LastRow
27             PicPath = PicFolder & .Range("B" & ItemRow).Value 'Full Picture Path
28             If Dir(PicPath, vbDirectory) = "" Or PicFolder = "\" Then GoTo NoPic
29             .Pictures.Insert(PicPath).Name = "ItemPic" & ItemRow 'Unique picture name
30             With .Shapes("ItemPic" & ItemRow)
31                 .Left = InvWkSht.Range("K" & ItemRow).Left + 10 'Set Left Postion
32                 .Top = InvWkSht.Range("K" & ItemRow).Top
33                 .LockAspectRatio = msoCTrue 'Lock Aspect Ration
34                 .Height = InvWkSht.Range("K" & ItemRow).Height - 2 'Set Picture height
35             End With
36             NoPic:
37         Next ItemRow
38     End With
39 End Sub
40
41 Sub InventoryAdj_Load()
42     With InvWkSht
43         If .Range("B4").Value = Empty Then
44             MsgBox "Please make sure to navigate to a correct adjustment"
45             Exit Sub
46         End If
47         If .Range("B2").Value = True Then
48             If MsgBox("Changes have been made to this adjustment. Are you sure you want
49             to clear these changes without saving?", vbYesNo, "Changes Not Saved") =
50             vbNo Then Exit Sub
51         End If
52         .Range("F4:G4,A7:O9999").ClearContents 'Clear All existing data
53         'Remove Existing Pictures
54         For Each PicShp In .Shapes
55             If InStr(PicShp.Name, "ItemPic") > 0 Then PicShp.Delete 'Delete picture
56             shape
57         Next PicShp
58         .Range("B2").Value = False
59     End With
60 End Sub

```

1

```

1
56 With AdjItems
57 LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row Data
58 If LastRow < 3 Then Exit Sub
59 .Range("A2:Gs" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
" H2:H3" ), CopyToRange:=.Range("J3:T3" ), Unique:=True
60 LastResultRow = .Range("J99999" ).End(xlUp).Row
61 If LastResultRow < 4 Then Exit Sub
62 .Range("K4:R" & LastResultRow).Formula = .Range("K2:R2" ).Formula 'Copy down
item formulas
63 InvWkSht.Range("A7:I" & LastResultRow + 3).Value = .Range("J4:R" &
LastResultRow).Value 'Items Details
64 InvWkSht.Range("J7:M" & LastResultRow + 3).Formula = InvWkSht.Range("J5:M5" ).
Formula 'Bring Over Bar code and Value Adjustment Formulas
65 InvWkSht.Range("L7:L" & LastResultRow + 3).Value = .Range("S4:S" &
LastResultRow).Value 'Actual Qty
66 InvWkSht.Range("O7:O" & LastResultRow + 3).Value = .Range("T4:T" &
LastResultRow).Value 'Adjustment Item Row
67 End With
68 InventoryAdj_AddPictures 'Add Pictures
69 End Sub
70 Sub InventoryAdj_LoadItems()
71 With InvWkSht
72 If .Range("B2" ).Value = True Then
73 If MsgBox("Changes have been made to this adjustment. Are you sure you want
to clear these changes without saving?" , vbYesNo, "Changes Not Saved" ) =
vbNo Then Exit Sub
74 End If
75 Application.ScreenUpdating = False
76 .Range("A7:O9999" ).ClearContents 'Clear All existing data
77 'Remove Existing Pictures
78 For Each PicShp In .Shapes
79 If InStr(PicShp.Name, "ItemPic" ) > 0 Then PicShp.Delete 'Delete picture
shape
80 Next PicShp
81 SortCol = .Range("B1" ).Value 'Set Sort Column
82 End With
83
84 With Items
85 LastRow = .Range("A99999" ).End(xlUp).Row 'Last Row
86 If LastRow < 3 Then GoTo NoData
87 .Range("A2:Q" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"S2:S3" ), CopyToRange:=.Range("AA2:AJ2" ), Unique:=True
88 LastResultRow = .Range("AA99999" ).End(xlUp).Row
89 If LastResultRow < 3 Then GoTo NoData
90
91 'Run Sort
92 With .Sort
93 .SortFields.Clear
94 .SortFields.Add Key:=Items.Cells(3, SortCol), SortOn:=xlSortOnValues, Order:=
xlAscending, DataOption:=xlSortNormal 'Sort Based on User entry
95 .SetRange Items.Range("AA3:AJ" & LastResultRow)
96 .Apply
97 End With
98 InvWkSht.Range("A7:I" & LastResultRow + 4).Value = .Range("AA3:AI" &
LastResultRow).Value 'Bring Over Item Data
99 InvWkSht.Range("N7:N" & LastResultRow + 4).Value = .Range("AJ3:AJ" &
LastResultRow).Value 'Bring over Item DB Row
100 End With
101
102 With InvWkSht

```

1 2

```

1 2
103 LastRow = LastResultRow + 4 'Last Row
104 .Range("J7:M" & LastRow).Formula = .Range("J5:M5" ).Formula
105 InventoryAdj_AddPictures 'Add Item Pictures
106 End With
107 NoData:
108 Application.ScreenUpdating = True
109
110 End Sub
111
112 Sub InventoryAdj_New()
113 With InvWkSht
114 If .Range("B2" ).Value = True Then
115 If MsgBox("Changes have been made to this adjustment. Are you sure you want
to clear these changes without saving?" , vbYesNo, "Changes Not Saved" ) =
vbNo Then Exit Sub
116 End If
117 .Range("B3,F4:G4,A7:O9999" ).ClearContents 'Clear All existing data
118 'Remove Existing Pictures
119 For Each PicShp In .Shapes
120 If InStr(PicShp.Name, "ItemPic" ) > 0 Then PicShp.Delete 'Delete picture
shape
121 Next PicShp
122 .Range("B2" ).Value = False
123 InventoryAdj_LoadItems
124 End With
125 End Sub
126
127 Sub InventoryAdj_Save()
128 With InvWkSht
129 If Application.WorksheetFunction.CountA(.Range("L7:L9999" )) < 1 Then
130 MsgBox "Please make at least one Inventory Adjustment before saving"
131 Exit Sub
132 End If
133 If .Range("B4" ).Value = Empty Then 'New Adjustmen
134 AdjRow = Adjust.Range("A9999" ).End(xlUp).Row + 1 'First avail Row
135 .Range("B3" ).Value = .Range("B5" ).Value 'Set New Adj. ID
136 Adjust.Range("A" & AdjRow).Value = .Range("B5" ).Value 'New Adj ID
137 Else 'Existing
138 AdjRow = .Range("B4" ).Value 'Exiting Adj. Row
139 End If
140 Adjust.Range("B" & AdjRow).Value = Date 'Set Current Date
141 Adjust.Range("C" & AdjRow).Value = .Range("M4" ).Value 'Total Adjusted Value
142
'Add Adjusted Items
143 LastRow = .Range("A9999" ).End(xlUp).Row
144 For ItemRow = 7 To LastRow
145 If .Range("L" & ItemRow).Value <> Empty Then 'Item Qty Has been Adjusted
146 ItemDBRow = .Range("N" & ItemRow).Value 'Item DB Row
147 If ItemDBRow <> 0 Then Items.Range("M" & ItemDBRow).Value = .Range("L" &
ItemRow).Value 'Update New Qty On Hand
148 If .Range("O" & ItemRow).Value = Empty Then 'New Adjustment Item
149 AdjDBRow = AdjItems.Range("A9999" ).End(xlUp).Row + 1 'First avail Row
150 AdjItems.Range("A" & AdjDBRow).Value = .Range("B3" ).Value 'Set
Adjustment ID
151 AdjItems.Range("F" & AdjDBRow).Value = "=row()" 'Set Formula for row
#
152 .Range("O" & ItemRow).Value = AdjDBRow 'Set Database Row
153 Else 'Existing Adjustment Item
154 AdjDBRow = .Range("O" & ItemRow).Value 'Existing DB Row

```

```

156     End If
157     AdjItems.Range("B" & AdjDBRow).Value = Date 'Set Current Date
158     AdjItems.Range("C" & AdjDBRow).Value = .Range("A" & ItemRow).Value 'Add
Item ID
159     AdjItems.Range("D" & AdjDBRow).Value = .Range("L" & ItemRow).Value
'Actual Qty\
160     AdjItems.Range("E" & AdjDBRow).Value = .Range("L" & ItemRow).Value - .
Range("I" & ItemRow).Value 'Adjusted Qty
161     End If
162     Next ItemRow
163     End With
164     End Sub
165
166     Sub NextAdj()
167         With InvWkSht
168             AdjID = .Range("B3" ).Value 'Adjustment ID
169             If AdjID = 0 Then 'No Current ID
170                 If .Range("B5" ).Value = 1 Then 'No Saved Adjustments
171                     MsgBox "Please save any adjustments first before navigating to previously
saved"
172                     Exit Sub
173                 End If
174                 .Range("B3" ).Value = .Range("B3" ).Value + 1 'Set Adj. ID to the last one
created
175                 InventoryAdj_Load
176                 Exit Sub
177             End If
178             If AdjID = .Range("B5" ).Value - 1 Then
179                 MsgBox "You are are already at the last adjustment created"
180                 Exit Sub
181             End If
182             .Range("B3" ).Value = .Range("B3" ).Value + 1 'Set next Adjust ID
183             InventoryAdj_Load
184         End With
185     End Sub
186
187     Sub PrevAdj()
188         With InvWkSht
189             AdjID = .Range("B3" ).Value 'Adjustment ID
190             If AdjID = 0 Then 'No Current ID
191                 If .Range("B5" ).Value = 1 Then 'No Saved Adjustments
192                     MsgBox "Please save any adjustments first before navigating to previously
saved"
193                     Exit Sub
194                 End If
195                 .Range("B3" ).Value = .Range("B5" ).Value - 1 'Set Adj. ID to the last one
created
196                 InventoryAdj_Load
197                 Exit Sub
198             End If
199             If AdjID = 1 Then
200                 MsgBox "You are are already at the first adjustment created"
201                 Exit Sub
202             End If
203             .Range("B3" ).Value = .Range("B3" ).Value - 1 'Set Previous Adjust ID
204             InventoryAdj_Load
205         End With
206     End Sub
207
208     Sub Print_InventoryWkSht()
1

```

```
1
209 With InvWkSht
210     LastRow = .Range("A99999" ).End(xlUp).Row
211     If LastRow < 7 Then
212         MsgBox "Please make sure to add items before printing the worksheet"
213         Exit Sub
214     End If
215     .PageSetup.PrintArea = "$D$5:$L$" & LastRow
216     .PrintOut , , , False, True, False, , , False
217
218 End With
219 End Sub
220
221 Sub ShowAdjHistory()
222     With InvWkSht.Shapes("AdjusHistoryPopUp" )
223         .Left = Range("D" & ActiveCell.Row).Left
224         .Top = Range("D" & ActiveCell.Row + 1).Top
225         .Visible = msoCTrue
226     End With
227 End Sub
```

A

ActiveCell, 11
 Add, 8
 AdjDBRow, 7, 9, 10
 AdjID, 7, 10
 AdjItems, 4, 7-10
 AdjRow, 7, 9
 Adjust, 9
 AdjustItemHistory, 4, 7
 AdvancedFilter, 7, 8
 Application, 8, 9
 Apply, 8

C

Cells, 8
 Clear, 8
 ClearContents, 7-9
 CopyToRange, 7, 8
 CountA, 9
 CountLarge, 4
 CriteriaRange, 7, 8

D

DataOption, 8
 Delete, 7-9
 Dir, 7

E

Empty, 4, 7, 9
 Explicit, 2-7

F

Formula, 8, 9

H

Height, 7

I

Insert, 7
 InStr, 7-9
 Intersect, 4
 InvAdj_ClearFilter, 7
 InventoryAdj_AddPictures, 7-9
 InventoryAdj_Load, 7, 10
 InventoryAdj_LoadItems, 4, 7-9
 InventoryAdj_New, 9
 InventoryAdj_Save, 9
 InvWkSht, 7-11
 ItemDBRow, 7, 9
 ItemRow, 7, 9, 10
 Items, 8, 9

K

Key, 8

L

LastResultRow, 7-9
 LastRow, 7-9, 11
 Left, 4, 7, 11
 LockAspectRatio, 7

M

MsgBox, 7-11
 msoCTrue, 4, 7, 11
 msoFalse, 4

N

Name, 7-9
 NextAdj, 10

NoData, 8, 9
 NoPic, 7

O

Order, 8

P

PageSetup, 11
 PicFolder, 7
 PicPath, 7
 PicShp, 7-9
 Pictures, 7
 PrevAdj, 10
 Print_InventoryWkSht, 10
 PrintArea, 11
 PrintOut, 11

R

Range, 4, 7-11
 Row, 4, 7-9, 11

S

ScreenUpdating, 8, 9
 SetRange, 8
 Shape, 7
 Shapes, 4, 7-9, 11
 ShowAdjHistory, 11
 Sort, 8
 SortCol, 7, 8
 SortFields, 8
 SortOn, 8

T

Target, 4
 Top, 4, 7, 11

U

Unique, 7, 8

V

Value, 4, 7-10
 vbDirectory, 7
 vbNo, 7-9
 vbYesNo, 7-9
 Visible, 4, 11

W

Worksheet_Change, 4
 Worksheet_SelectionChange, 4
 WorksheetFunction, 9

X

xlAscending, 8
 xlFilterCopy, 7, 8
 xlSortNormal, 8
 xlSortOnValues, 8
 xlUp, 7-9, 11

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.



Excel For Freelancers