

```

1 Function Pivot_copy(sPivotName As String, wsPivot As Worksheet, wsTarget As Worksheet,
2   lStartRow As Long) As Long
3   'Sub Pivot_copy()'Sub
4   'Dim pvtResourceSplitOnsite As PivotTable
5   'Dim wbNew As Workbook 'Sub
6   'Dim wsNew As Worksheet 'Sub
7   Dim rFirstLabel, rLastLabel, rTotal, rTableRange As Range
8   'Dim lTotalsRow, lRateCol, lStartCol, lStartRow, lOffset As Long 'Sub
9   Dim lTotalsRow, lRateCol, lStartCol, lOffset As Long
10  Dim cSum As Currency
11  'Dim sPivotName, sBillingFilter As String 'Sub
12  'sPivotName = "Parent Task Split"'Sub
13  ''sPivotName = "Resource Split"
14  '
15  'sBillingFilter = "LIFT"'Sub
16
17  'Source
18  'With ActiveSheet.PivotTables(sPivotName) 'sub
19  With wsPivot.PivotTables(sPivotName)
20    'Refresh the Pivot fromm source
21    '.RefreshDataSourcesValues
22    '.PivotFields("LIFT/RUN").CurrentPage = sBillingFilter
23    Set rTableRange = .TableRange1
24    With .TableRange1
25
26      If sPivotName = "Component Split" Then
27        Set rFirstLabel = .Cells.Find(What:="Row Labels", After:=.Cells(1), _
28          LookAt:=xlWhole, SearchDirection:=xlNext)
29        Set rLastLabel = .Cells.Find(What:="% of Cost", After:=.Cells(1), _
30          LookAt:=xlWhole, SearchDirection:=xlNext)
31      Else
32        Set rFirstLabel = .Cells.Find(What:="Project Role", After:=.Cells(1), _
33          LookAt:=xlWhole, SearchDirection:=xlNext)
34        Set rLastLabel = .Cells.Find(What:="Sum Of Cost", After:=.Cells(1), _
35          LookAt:=xlWhole, SearchDirection:=xlNext)
36      End If
37      ' Total of rows
38      Set rTotal = .Cells.Find(What:="Grand Total", After:=.Cells(1), _
39        LookAt:=xlWhole, SearchDirection:=xlNext)
40    End With
41
42  End With
43  lTotalsRow = rTotal.Row
44
45
46  ''Destination
47  'Set wbNew = Workbooks.Add 'Sub
48  ''Set wsNew = wbNew.Worksheet.Add()
49  '
50  'With ActiveSheet'Sub
51
52  With wsTarget 'Function
53    'Set rStartingPoint = .Cells(1, 1)
54    lStartCol = 2
55    'lStartRow = 2 'sub
56

```

```

57 If sPivotName = "Component Split" Then
58     lOffset = 3
59     rTableRange.Copy (.Cells(lStartRow, lStartCol + lOffset))
60
61     .Cells(lStartRow + 1, lStartCol + lOffset).Resize(, 5).Value = _
62     Array("Issue summary", "Effort (Hr)", "Effort %", "Cost", "Cost %")
63     lStartCol = lStartCol + lOffset
64     lOffset = 0
65 Else
66     rTableRange.Copy (.Cells(lStartRow, lStartCol - 1))
67
68     .Cells(lStartRow + 1, lStartCol - 1).Resize(, 7).Value = _
69     Array("Project Role", "Resource Name", "Location", "Effort (Hr)", "Cost",
70     "Out of office/Vacations", "Holidays")
71     lOffset = 3
72     lStartCol = lStartCol - 1
73 End If
74
75     .Cells(lStartRow, lStartCol + lOffset + 1).Value = ""
76 With .Range(Cells(lStartRow, lStartCol), _
77     Cells(lStartRow, lStartCol + rLastLabel.Column - rFirstLabel.Column +
78     lOffset)).Interior
79     .Pattern = xlNone
80     .TintAndShade = 0
81     .PatternTintAndShade = 0
82 End With
83 lStartRow = lStartRow + 1
84
85 With .Range(.Cells(lStartRow, lStartCol), _
86     .Cells(lStartRow, lStartCol + rLastLabel.Column - rFirstLabel.Column + lOffset
87     ))
88     .Font.Bold = True
89     With .Interior
90         .Pattern = xlSolid
91         .PatternColorIndex = xlAutomatic
92         .Color = RGB(198, 224, 180)
93         .TintAndShade = 0
94         .PatternTintAndShade = 0
95     End With
96 End With
97
98 ' All borders
99 With .Range(.Cells(lStartRow, lStartCol), _
100     .Cells(rTotal.Row - rFirstLabel.Row + lStartRow, lStartCol + rLastLabel.Column
101     - rFirstLabel.Column + lOffset))
102     With .Borders
103         .LineStyle = xlContinous
104         .Color = vbBlack
105         .Weight = xlThin
106     End With
107 End With
108
109 'Total
110 With .Range(Cells(rTotal.Row - rFirstLabel.Row + lStartRow, lStartCol), _
111     Cells(rTotal.Row - rFirstLabel.Row + lStartRow, lStartCol + rLastLabel.Column
112     - rFirstLabel.Column + lOffset))
113     .Interior.ColorIndex = 0
114     With .Borders(xlEdgeTop)

```

```

110     .LineStyle = xlContinuous
111     .ColorIndex = 0
112     .TintAndShade = 0
113     .Weight = xlMedium
114   End With
115   With .Borders(xlEdgeBottom)
116     .LineStyle = xlContinuous
117     .ColorIndex = 0
118     .TintAndShade = 0
119     .Weight = xlMedium
120   End With
121 
122   End With
123 
124 End With
125 'Pivot_copy = rTotal.Row
126 Pivot_copy = rTotal.Row - rFirstLabel.Row + lStartRow
127 'Removing pivot filter
128 ' wsPivot.PivotTables(sPivotName).PivotFields("LIFT/RUN").CurrentPage = "All"
129 End Function
130 Function log(rLog As Range, sMessage As String) As Range
131 
132   rLog.Value = Now
133   rLog.Offset(, 1).Value = sMessage
134   Set log = rLog.Offset(1, 0)
135 
136 End Function
137 
138 
139 Function Pivot_copy_Resources_n_task(sPivotName As String, wsPivot As Worksheet,
140 wsTarget As Worksheet, lStartRow As Long) As Long
141 
142 'Sub Pivot_copy_Resources_n_task() 'Sub
143 'Dim pvtResourceSplitOnsite As PivotTable 'sub
144 'Dim wbNew As Workbook 'sub
145 'Dim wsNew As Worksheet 'sub
146 Dim rFullscreen, rRate, rTotal, rNamesNTasks, rTotalCol, rPivotTotal, rDataRange,
147 rStartingPoint As Range
148 'Dim lTotalsRow, lRateCol, lStartCol, lStartRow As Long 'sub
149 Dim lTotalsRow, lRateCol, lStartCol As Long
150 Dim cSum As Currency
151 'Dim sPivotName, sBillingFilter As String 'sub
152 
153 'sPivotName = "Resource and Task Split Onshore" 'sub
154 'sBillingFilter = "LIFT" 'sub
155 
156 'Source
157 'With ActiveSheet.PivotTables(sPivotName) 'sub
158 With wsPivot.PivotTables(sPivotName)
159   'Refresh the Pivot from source
160   '.RefreshDataSourceValues
161   '.PivotFields("LIFT/RUN").CurrentPage = sBillingFilter
162   With .RowRange
163     'Finds beginning and end coordinates
164     Set rFullscreen = .Cells.Find(What:="Full name", After:=.Cells(1),
165       LookAt:=xlWhole, SearchDirection:=xlNext)
166     Set rRate = .Cells.Find(What:="Rate", After:=.Cells(1),

```

```

166             LookAt:=xlWhole, SearchDirection:=xlNext)
167             ' Total of rows
168             Set rTotal = .Cells.Find(What:="Grand Total", After:=.Cells(1), _
169                 LookAt:=xlWhole, SearchDirection:=xlNext)
170             'Set rTotalCol = .Range(.Cells(rFullscreen.Row, 1), .Cells(rFullscreen.Row,_
171                 1000)).Find(What:="Grand Total", After:=.Cells(1), _
172                 LookAt:=xlWhole, SearchDirection:=xlNext)
173         End With
174         With .ColumnRange
175             'total of columns
176             Set rTotalCol = .Cells.Find(What:="Grand Total", After:=.Cells(1), _
177                 LookAt:=xlWhole, SearchOrder:=xlRows, SearchDirection:=xlNext)
178         End With
179     lTotalsRow = rTotal.Row
180     lRateCol = rRate.Column
181
182     'Selecting the cells from "Full name" to last row in Rate Column
183     Set rNamesNTasks = wsPivot.Range(wsPivot.Cells(rFullscreen.Row, rFullscreen.Column),_
wsPivot.Cells(lTotalsRow, lRateCol))
184     'Selecting whole Grand Total column
185     Set rPivotTotal = wsPivot.Range(wsPivot.Cells(rTotalCol.Row, rTotalCol.Column),_
wsPivot.Cells(lTotalsRow, rTotalCol.Column))
186     'Selecting all the daily hours cells
187     Set rDataRange = wsPivot.Range(wsPivot.Cells(rTotalCol.Row, rRate.Column + 1),_
wsPivot.Cells(rTotal.Row, rTotalCol.Column - 1))
188
189     ''Destination
190     'Set wbNew = Workbooks.Add
191     ''Set wsNew = wbNew.Worksheet.Add()
192
193     'With ActiveSheet
194
195     With wsTarget
196         'Set rStartingPoint = .Cells(1, 1)
197         lStartCol = 2
198         'lStartRow = 1 'sub
199
200         'Onshore/offshore title
201         With .Cells(lStartRow - 1, lStartCol)
202             If InStr(1, sPivotName, "Onshore", vbTextCompare) > 0 Then
203                 .Value = "Onshore"
204             Else
205                 .Value = "Offshore"
206             End If
207             .Font.Bold = True
208             With .Interior
209                 .Pattern = xlSolid
210                 .PatternColorIndex = xlAutomatic
211                 .Color = RGB(159, 194, 230)
212                 .TintAndShade = 0
213                 .PatternTintAndShade = 0
214             End With
215         End With
216
217         rNamesNTasks.Copy (.Cells(lStartRow, lStartCol))
218         rPivotTotal.Copy (.Cells(lStartRow, lStartCol + lRateCol - rFullscreen.Column + 1))
219         rDataRange.Copy (.Cells(lStartRow, lStartCol + lRateCol - rFullscreen.Column + 3))

```

```

220
221     .Cells(lStartRow, lStartCol + lRateCol - rFullscreen.Column + 2).Value = "Cost"
222     .Cells(lStartRow, lStartCol + lRateCol - rFullscreen.Column + 1).Value = "Effort"
223     .Cells(lStartRow, lStartCol + lRateCol - 2).Value = "Project Name"
224
225     For Each c In .Range(.Cells(lStartRow + 1, lStartCol + lRateCol - rFullscreen.Column
226     + 2), _
227         .Cells(lStartRow + lTotalsRow - rFullscreen.Row, lStartCol + lRateCol - rFullscreen.Column + 2))
228         'Cost=Effort*rate
229         'c.NumberFormat = "$#,##0.00"
230         'Accounting number format
231         c.NumberFormat = "_($* #,##0.00_);_($* (#,##0.00);_($* ""_??_);_(@_)"
232         c.Value = .Cells(c.Row, c.Column - 1).Value * .Cells(c.Row, c.Column - 2).
233         Value
234
235         cSum = c.Value + cSum
236     Next c
237
238     With .Cells(lStartRow + lTotalsRow - rFullscreen.Row, lStartCol + lRateCol - rFullscreen.Column + 2)
239         .Value = cSum
240         .Font.Bold = True
241         .NumberFormat = "_($* #,##0.00_);_($* (#,##0.00);_($* ""_??_);_(@_)"
242     End With
243
244     With .Range(.Cells(lStartRow, lStartCol), Cells(lStartRow, lStartCol +
245     rTotalCol.Column - rFullscreen.Column + 1))
246         .Font.Bold = True
247         With .Interior
248             .Pattern = xlSolid
249             .PatternColorIndex = xlAutomatic
250             .Color = RGB(198, 224, 180)
251             .TintAndShade = 0
252             .PatternTintAndShade = 0
253         End With
254     End With
255
256     'All borders
257     With .Range(.Cells(lStartRow, lStartCol),
258         Cells(rTotal.Row - rFullscreen.Row + lStartRow, lStartCol + rTotalCol.Column - rFullscreen.Column + 1))
259         With .Borders
260             .LineStyle = xlContinous
261             .Color = vbBlack
262             .Weight = xlThin
263         End With
264     End With
265
266     'Names
267     With .Range(.Cells(lStartRow + 1, lStartCol),
268         Cells(rTotal.Row - rFullscreen.Row + lStartRow - 1, lStartCol))
269         .Interior.ColorIndex = 0
270         With .Borders(xlInsideHorizontal)
271             .LineStyle = xlContinous
272             .Color = vbWhite
273             .Weight = xlThin

```

```

272     End With
273 End With
274
275 'Total
276 With .Range(Cells(rTotal.Row - rFullscreen.Row + lStartRow, lStartCol),
277 Cells(rTotal.Row - rFullscreen.Row + lStartRow, lStartCol + rTotalCol.Column -
278 rFullscreen.Column + 1))
279     .Interior.ColorIndex = 0
280     With .Borders(xlEdgeTop)
281         .LineStyle = xlContinuous
282         .ColorIndex = 0
283         .TintAndShade = 0
284         .Weight = xlMedium
285     End With
286     With .Borders(xlEdgeBottom)
287         .LineStyle = xlContinuous
288         .ColorIndex = 0
289         .TintAndShade = 0
290         .Weight = xlMedium
291     End With
292 End With
293
294 End With
295 Pivot_copy_Resources_n_task = rTotal.Row - rFullscreen.Row + lStartRow
296 'Pivot_copy_Resources_n_task = rTotal.Row
297 'Removing pivot filter
298 'wsPivot.PivotTables(sPivotName).PivotFields("LIFT/RUN").CurrentPage = "All"
299 End Function
300
301
302 Sub Prepare_sheet()
303
304 'Dim fso As New Scripting.FileSystemObject
305 Dim vPath, vWCPath As Variant
306 Dim wBook, wBookNew As Workbook
307 Dim wsPivot As Worksheet
308 Dim wsTarget As Worksheet
309 Dim wsLog As Worksheet
310 Dim sSourcefile, sWorkFile, sPivotName As String
311 Dim lResult As Long
312 Dim fso As New Scripting.FileSystemObject
313 Dim rLog As Range
314
315
316
317 vPath = ChooseFile("Choose the File to be prepared")
318 Set wBook = Workbooks.Open(vPath)
319 Set wsPivot = wBook.Sheets("Pivots")
320 Set wsLog = wBook.Sheets.Add
321 wsLog.Name = wsLog.Name + " Log"
322 Set rLog = wsLog.Cells(1, 1)
323 Set rLog = log(rLog, "starting the loop") 'log
324
325
326
327 'For Each c In Array("Lift", "Run")
328     'Adding worksheet for Summary

```

```

329     rLog = log(rLog, "Looping for " + c) 'log
330     Set wsTarget = wBook.Sheets.Add
331     Set rLog = log(rLog, "creating sheet " + wsTarget.Name) 'log
332     wsTarget.Name = wsTarget.Name
333
334     sPivotName = "Resource and Task Split Onshore"
335     Set rLog = log(rLog, "calling Pivot_copy_Resources_n_task(" + sPivotName + ", " +
336     wsPivot.Name + ", " + wsTarget.Name + ", 3)") 'log
337     lResult = Pivot_copy_Resources_n_task(sPivotName, wsPivot, wsTarget, 3)
338     Set rLog = log(rLog, "Returned last row as " + Str(lResult)) 'log
339
340     sPivotName = "Resource and Task Split Offshore"
341     Set rLog = log(rLog, "calling Pivot_copy_Resources_n_task(" + sPivotName + ", " +
342     wsPivot.Name + ", " + wsTarget.Name + ", " + Str(lResult + 6) + ")") 'log
343     lResult = Pivot_copy_Resources_n_task(sPivotName, wsPivot, wsTarget, lResult + 6)
344     Set rLog = log(rLog, "Returned last row as " + Str(lResult)) 'log
345
346     sPivotName = "Component Split"
347     Set rLog = log(rLog, "calling Pivot_copy(" + sPivotName + ", " + wsPivot.Name +
348     ", " + wsTarget.Name + ", " + Str(lResult + 6) + ")") 'log
349     lResult = Pivot_copy(sPivotName, wsPivot, wsTarget, lResult + 6)
350     Set rLog = log(rLog, "Returned last row as " + Str(lResult)) 'log
351
352 'Next c
353
354
355
356 End Sub
357 Function ChooseFile(sTitle As String) As Variant
358
359
360 Dim fd As FileDialog
361
362
363
364 Set fd = Application.FileDialog(msoFileDialogFilePicker)
365
366
367
368 'the number of the button chosen
369
370 Dim FileChosen As Integer
371
372
373
374
375
376 '1) To set the caption of the dialog box,
377
378 ' set the Title property
379
380 fd.Title = sTitle '"Choose Billing File downloaded"
381
382 FileChosen = fd.Show

```

```
383  
384 '2) Set the oddly named InitialFileName property to  
385 ' determine the initial folder selected  
386 'fd.InitialFileName = "c:\wise owl\"  
387  
388 '3) Set the InitialView property to control how your files  
389 ' appear on screen (as a list, icons, etc.)  
390 fd.InitialView = msoFileDialogViewSmallIcons  
391  
392 '4) To set the filters (you can have as many as you like)  
393 ' first clear any existing ones, then add them one by one  
394 'fd.Filters.Clear  
395  
396 'fd.Filters.Add "Excel macros", "*.xlsm"  
397  
398 ' if there's more than one filter, you can control which  
399 ' one is selected by default  
400 'fd.FilterIndex = 1  
401  
402 '5) Set the ButtonName property to control the text on  
403 ' the OK button (the ampersand means the following  
404 ' letter is underlined and choosable with the ALT key)  
405 'fd.ButtonName = "Choose &owly file"  
406  
407  
408 If FileChosen <> -1 Then  
409  
410  
411 'didn't choose anything (clicked on CANCEL)  
412 MsgBox "You chose cancel"  
413  
414  
415 Else  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440
```

```
*441 'display name and path of file chosen
442 fd.Title = "File selection dialog"
443
444 MsgBox fd.SelectedItems(1)
445
446 ChooseFile = fd.SelectedItems(1)
447
448 End If
449
450
451
452
453
454
455 End Function
456
457
```

```

With .PivotFields("% of Effort")
    .Calculation = xlPercentOfTotal
    .NumberFormat = "0.00%"
End With

With .PivotFields("% of cost")
    .Calculation = xlPercentOfTotal
    .NumberFormat = "0.00%"
End With

'Applying Filter
With .PivotFields("Issue Summary")
    .PivotItems("Out of Office").Visible = False
    '.PivotItems("Out of the office").Visible = False
End With

With .PivotFields("Project Name")
    On Error Resume Next
    .PivotItems("Digital Strategy").Visible = False
    On Error GoTo 0
    '.PivotItems("Out of the office").Visible = False
End With

End With

iLastRow = LastRow(wsPivotSheet)

'resource split
'Where do you want Pivot Table to start?
sResourceSplit = .Cells(10 + iLastRow, 1).Address(ReferenceStyle:=xlR1C1)

'Create Pivot table from Pivot Cache
Set pvtResourceSplit = pvtCache.CreatePivotTable(
TableDestination:=sResourceSplit,
TableName:="Resource Split")

With pvtResourceSplit
    .PivotFields("Project Name").Orientation = xlPageField ' Filter for CDS vs
EDW Project

    .PivotFields("Project Role").Orientation = xlRowField
    .PivotFields("Full Name").Orientation = xlRowField
    .PivotFields("Location").Orientation = xlRowField

    .PivotFields("Issue Summary").Orientation = xlPageField

    .AddDataField .PivotFields("Hours"), "Effort", xlSum
    .AddDataField .PivotFields("Cost"), "Sum of Cost", xlSum

    'Formatting
    .PivotFields("Sum of Cost").NumberFormat = "$* #,##0.00"; "$* (#,##0.00); $* ""-""??_);_(@_)" ' Accounting Format

    'Selecting tabular layout vs outline
    .RowAxisLayout xlTabularRow
    ' Repeate labels
    .RepeatAllLabels xlRepeatLabels
    'Remove +/- button

```

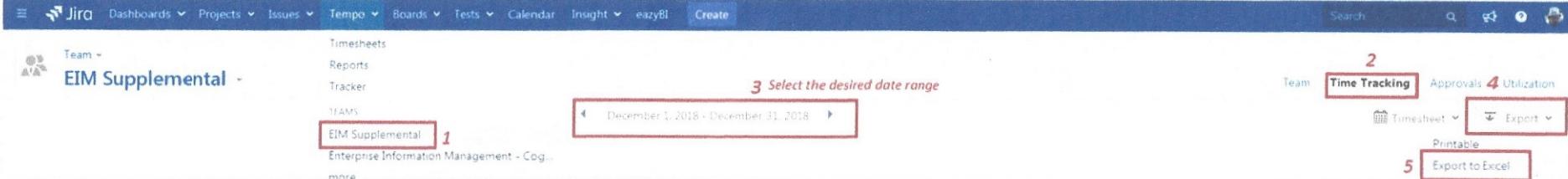
```

1 .ShowDrillIndicators = False
2
3     'Sets Sub total to None for each row label
4     .PivotFields("Project Role").Subtotals = Array(False, False, False, False,
5     False, False, False, False, False, False, False, False)
6     .PivotFields("Location").Subtotals = Array(False, False, False, False, False,
7     False, False, False, False, False, False, False)
8     .PivotFields("Full Name").Subtotals = Array(False, False, False, False, False,
9     False, False, False, False, False, False, False)
' .PivotField("Location").AutoSort xlDescending
10
11
12     'Applying Filter
13     With .PivotFields("Project Name")
14         On Error Resume Next
15             .PivotItems("Digital Strategy").Visible = False
16             '.PivotItems("Out of the office").Visible = False
17             On Error GoTo 0
18         End With
19
20     End With
21
22     'Category split
23     'Where do you want Pivot Table to start?
24     sCategorySplit = .Cells(10 + iLastRow, 10).Address(ReferenceStyle:=xlR1C1)
25
26     'Create Pivot table from Pivot Cache
27     Set pvtCategorySplit = pvtCache.CreatePivotTable( _
28         TableDestination:=sCategorySplit,
29         TableName:="Project Category Split")
30
31     'Formatting the pivot tables
32
33     On Error Resume Next
34     With pvtResourceSplit.PivotFields("Issue Summary")
35         .PivotItems("Out of Office").Visible = False
36         .PivotItems("Out of the office").Visible = False
37     End With
38
39     On Error GoTo 0
40
41     End With
42
43     'Closing rate sheet
44     wbRate.Close
45
46     End Sub
47
48     Sub MonthendSummary()
49
50         Dim sURLBase, sRateSheetPath As String
51         Dim sStartDate, sEndDate, sResult, sParentTaskReportURL As String
52         Dim vPath As Variant
53         Dim wbWorking As Workbook
54
55         'Picking up the Ratesheet path
56         sRateSheetPath = ActiveSheet.Cells(4, 3).Text()
57
58         'Create working copy of the billing sheet
59         Set wbWorking = makeWorkingCopy
60         'Create Pivot

```

||

```
325 Call UpdateWorklog(wbWorking, sRateSheetPath)
326
327 End Sub
328
329
```

A	B	C	D
<b>Instructions to generate report</b>			
<b>1 Rate Sheet</b>	Rate sheet is a look up sheet with information like, Hourly rate, Project Role, Technology, Location and Cognizant ID. Update the complete path to that sheet here <i>C:\Binil\Test\Jira_auto\Rates_V13.0.xlsx</i>		
<b>2 JIRA report</b>	Login to <a href="https://jira.alfains.com">https://jira.alfains.com</a> with a user who has access to all necessary user timesheets. Follow the steps shown to generate a tempo timesheet report and save to local folder(Preferably an empty folder)		
<b>3 Generate Pivot</b>			
<b>4 Select Report prompt</b>	The script will prompt the user to select the downloaded report file		
<b>5 Parent task</b>	The script will make a working copy of the downloaded report in xlsx format.		
<b>6 Pivot sheet</b>	The script will add rate sheet into the working copy.Then it will add a Pivot sheet and generate all the pivots.		
<b>7 Script ends</b>	At this point the script ends. The enhancements on working copy will not be saved. User can verify the data and choose to save this		
<b>8 Generate Report</b>	The final Billing report, has to be generated, out of the Pivot sheet, by copying the pivots in to separate sheets and formatting them. These steps are automated to an extent using another script which can be run by pressing the Button "Format Report"		
<b>9 Final touches</b>	Week number in the month need to be added to Onsite and Offshore effort tables . Any other formatting can be added too		

A B C D E F G H I J K

This sheet use a vba module named Jira\_report. Please enable

1 macros in trust center or from the warning ribbon

This macro needs some libraries to be enabled as well. Please

2 follow the steps

a Please press ALT+F11 to go to VBA editor

Please go to the menu bar Tools>References and enable the

b libraries as shown in the screenshot

3 Please save the rates sheet somewhere accessible.

4 Now the report can be generated from "Generate report" tab

References - VBAProject

✖

OK

Cancel

Browse...

Priority

Help

- Visual Basic For Applications
- Microsoft Excel 15.0 Object Library
- OLE Automation
- Microsoft Office 15.0 Object Library
- Microsoft Internet Controls
- Microsoft Scripting Runtime
- VBAProject
- AccessibilityCplAdmin 1.0 Type Library
- Acresso FLEXnet Connect Agent 11.5 Type Library
- Acresso FLEXnet Connect Download Manager 11.5 T
- Acresso FLEXnet Connect Software Manager 11.5 Ty
- Acresso FLEXnet Connect WebAgent 11.5 Type Libra
- Acrobat Access 3.0 Type Library
- AcroBroker1 ih

SBEserver 1.0 Type Library

Location: C:\Windows\ehome\CreateDisc\SBEserver.exe

Language: Standard