

```
1 Imports System.IO
2 Imports System.Text.RegularExpressions
3 Imports Microsoft.VisualBasic

4 Module Module1
5     Sub Main()
6         Dim currpath As String = Directory.GetCurrentDirectory
7         'remove after debugging
8         'currpath = "P:\etest\work\29149qd"
9         'currpath = "P:\etest\work\29756qd"
10        'currpath = "c:\etest\flex1"
11        'currpath = "C:\etest\test3"

12        'If job exists, give a warning!
13        Dim ToolNo As String = GetToolNo(currpath)
14        Dim JobName As String = currpath & "\" & ToolNo & ".net.bin"
15        If File.Exists(JobName) Then
16            Shell("c:\Pental~1\CAMMas~1\CAMMas~1.exe " & JobName, AppWinStyle.
                MinimizedNoFocus, False)
17            'MessageBox.Show("The Job already exists!", "Job Found!", _
18                '            MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
19            GoTo LastLine
20        End If

21        'Find CPL
22        Dim cpl As String = GetSS(currpath, "")

23        'Load Files
24        GetLayers(currpath, cpl)

25        'Cleanup
26        Dim cmpfile As String = currpath & "\compare.txt"
27        Dim fi As New FileInfo(cmpfile)
28        Console.WriteLine(fi)
29        If fi.Exists Then fi.Delete()
30        'That's all folks!
31        LastLine
32    End Sub

33    Public Function WriteToTempFile(ByVal Data As String) As String
34        Dim strFilename As String = Path.GetTempFileName()
35        Dim objFS As New System.IO.FileStream(strFilename, _
36            System.IO.FileMode.Append, _
37            System.IO.FileAccess.Write)
38        ' Opens stream and begins writing
39        Dim Writer As New StreamWriter(objFS)
40        Writer.BaseStream.Seek(0, SeekOrigin.End)
41        Writer.WriteLine(Data)
42        Writer.Flush()
43        ' Closes and returns temp path
44        Writer.Close()
45        Return strFilename
46    End Function

47    Function GetToolNo(ByVal currpath As String) As String
48        Dim myPath As String = currpath
49        Dim myDirectory As System.IO.DirectoryInfo
```

```

50     myDirectory = New System.IO.DirectoryInfo(myPath)
51     Dim ToolNo As String = Left(myDirectory.Name, 5)
52     Return ToolNo
53 End Function

54 Function GetSS(ByVal currpath As String, ByVal SolderSide As String) As String
55     Dim ssregex As String = "^\\w*[a-z]-[0-9]+[a-z]{1}\\b"
56     Dim options As RegexOptions = ((RegexOptions.IgnorePatternWhitespace _
57     Or RegexOptions.Multiline) _
58     Or RegexOptions.IgnoreCase)
59     Dim reg As Regex = New Regex(ssregex, options)

60     'Get directory list into sContent
61     Dim sContent
62     Dim oFile As File
63     Dim oReader As StreamReader
64     Dim oFileStream As FileStream
65     Dim oWriter As StreamWriter
66     Dim strFilename As String = WriteToTempFile("")
67     Dim file As String
68     Dim filename As String
69     Dim stripreg As String = "^.*\\"
70     Dim stripoptions As RegexOptions = ((RegexOptions.
71     IgnorePatternWhitespace _
72     Or RegexOptions.Multiline) _
73     Or RegexOptions.IgnoreCase)
74     Dim stripregex As Regex = New Regex(stripreg, stripoptions)

75     oFileStream = New FileStream(strFilename, FileMode.Create)
76     oWriter = New StreamWriter(oFileStream)

77     For Each file In Directory.GetFiles(currpath)
78         file = stripregex.Replace(file, stripreg, "")
79         'Remove after debugging
80         Console.WriteLine(file)
81         oWriter.WriteLine(file)
82     Next

83     oWriter.Close()
84     oFileStream.Close()
85     oReader = oFile.OpenText(strFilename)
86     sContent = oReader.ReadToEnd
87     oReader.Close()
88     oReader = Nothing
89     oFile = Nothing

90     Dim oMatches As MatchCollection
91     oMatches = reg.Matches(sContent)
92     SolderSide = Val(oMatches.Count)
93     Return SolderSide
94 End Function

95 Sub GetLayers(ByVal currpath As String, ByVal cpl As String)
96     Shell("c:\Pental~1\CAMMas~1\CAMMas~1.exe", AppWinStyle.MinimizedNoFocus)

```

```
        False)

96     Dim btImportJob As New CAMMaster.Tool
97     Dim lcount As Integer = 1
98     Dim lcolor As Integer = 1
99     Dim PlnPfx As String = "P0"
100    Dim PlnNum As Integer = 1
101    Dim ToolNo As String = GetToolNo(currpath)
102    Dim Fl0Name As String = ToolNo & "net."
103    Dim JobName As String = ToolNo & "net"

104    'Setup Expression
105    Dim lnum As Integer = 1
106    Dim layerregex As String = "^\\w*[a-z]-" & lnum & "[a-z]{1}\\b"
107    Dim options As RegexOptions = ((RegexOptions.IgnorePatternWhitespace _
108    Or RegexOptions.Multiline) _
109    Or RegexOptions.IgnoreCase)
110    Dim oRegex As Regex = New Regex(layerregex, options)
111    Dim oMatch As Match
112    Dim oMatches As MatchCollection

113    'Get directory list into sContent

114    Dim sContent
115    Dim oFile As File
116    Dim oReader As StreamReader
117    Dim oFileStream As FileStream
118    Dim oWriter As StreamWriter
119    Dim strFilename As String = WriteToTempFile("")
120    Dim file As String
121    Dim filename As String
122    Dim stripreg As String = "^.*\\\\"
123    Dim stripoptions As RegexOptions = ((RegexOptions.
124    IgnorePatternWhitespace _
125    Or RegexOptions.Multiline) _
126    Or RegexOptions.IgnoreCase)
127    Dim stripregex As Regex = New Regex(stripreg, stripoptions)

127    oFileStream = New FileStream(strFilename, FileMode.Create)
128    oWriter = New StreamWriter(oFileStream)

129    For Each file In Directory.GetFiles(currpath)
130        file = stripregex.Replace(file, stripreg, "")

131        'Remove after debugging
132        'Console.WriteLine(file)

133        oWriter.WriteLine(file)
134    Next

135    oWriter.Close()
136    oFileStream.Close()
137    oReader = oFile.OpenText(strFilename)
138    sContent = oReader.ReadToEnd
139    oReader.Close()
140    oReader = Nothing
```

```

141      oFile = Nothing

142      'get gerber layers
143      'kill screen updates in CAMMAster
144      btImportJob.MessagesMode = "Background"
145      oMatches = oRegex.Matches(sContent)
146      If oMatches.Count = 0 And lnum = 1 Then
147          Console.WriteLine("No Layer 1 Found")
148          Console.WriteLine("Press ENTER to QUIT")
149          Console.ReadLine()
150      End
151  End If

152  Do While oMatches.Count = 1
153      For Each oMatch In oMatches
154          btImportJob.CurrentDirectory = (currpath)
155          btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)

156          If lcount = 1 Then
157              btImportJob.LayerType(lcount) = "CPU"
158              btImportJob.LayerColor(lcount) = 12
159              If UCase$(Right$(oMatch.Value, 1)) = "N" Or UCase$(Right$(oMatch
160                  .Value, 1)) = "A" Then
161                  btImportJob.LayerNegative(lcount) = True
162                  'OldName = btImportJob.LayerName(lcount)
163                  btImportJob.LayerName(lcount) = Fl0Name + "1f"
164                  lcount = lcount + 1
165              Else
166                  btImportJob.LayerNegative(lcount) = False
167                  btImportJob.LayerColor(lcount) = 12
168                  btImportJob.LayerName(lcount) = Fl0Name + "1"
169                  lcount = lcount + 1
170              End If
171          ElseIf lcount = cpl Then
172              btImportJob.LayerType(lcount) = "CPL"
173              btImportJob.LayerColor(lcount) = 12
174              If UCase$(Right$(oMatch.Value, 1)) = "N" Or UCase$(Right$(oMatch
175                  .Value, 1)) = "A" Then
176                  btImportJob.LayerNegative(lcount) = True
177                  btImportJob.LayerName(lcount) = Fl0Name + Format(lcount) + "
178                      ssf"
179                  lcount = lcount + 1
180              Else
181                  btImportJob.LayerNegative(lcount) = False
182                  btImportJob.LayerColor(lcount) = 12
183                  btImportJob.LayerName(lcount) = Fl0Name + Format(lcount) + "ss
184                      "
185                  lcount = lcount + 1
186              End If
187          Else
188              Dim PlnFmt As String = Format(PlnNum)
189              If UCase$(Right$(oMatch.Value, 1)) = "N" Or UCase$(Right$(oMatch
190                  .Value, 1)) = "A" Then
191                  btImportJob.LayerNegative(lcount) = True
192                  If PlnNum > 9 Then PlnPfx = "P"
193                  btImportJob.LayerType(lcount) = PlnPfx + PlnFmt
194                  btImportJob.LayerColor(lcount) = lcolor

```

```

190         PlnNum = PlnNum + 1
191         btImportJob.LayerName(lcount) = Fl0Name + "g" + Format(lcount)
            + "f"
192         'btImportJob.LayerExtension(lcount) = ".g" + lcount + "f"
193     - ElseIf UCase$(Right$(oMatch.Value, 1)) = "P" Or UCase$(Right$(
            oMatch.Value, 1)) = "M" Then
194         btImportJob.LayerNegative(lcount) = False
195         If PlnNum > 9 Then PlnPfx = "P"
196         btImportJob.LayerType(lcount) = PlnPfx + PlnFmt
197         btImportJob.LayerColor(lcount) = lcolor
198         PlnNum = PlnNum + 1
199         btImportJob.LayerName(lcount) = Fl0Name + "g" + Format(lcount)
            + "d"
200         'btImportJob.LayerExtension(lcount) = ".g" + lcount + "d"
201     - Else
202         btImportJob.LayerNegative(lcount) = False
203         btImportJob.LayerType(lcount) = "SIG"
204         btImportJob.LayerName(lcount) = Fl0Name + Format(lcount)
205         'btImportJob.LayerExtension(lcount) = "." + lcount
206         btImportJob.LayerColor(lcount) = lcolor
207     - End If
208         lcount = lcount + 1
209     - End If

210     'Remove after debugging
211     'Console.WriteLine(oMatch.Value)

212 Next
213     lnum = lnum + 1
214     lcolor = lcolor + 1
215     layerregex = "^\\w*[a-z]-" & lnum & "[a-z]{1}\\b"
216     oRegex = New Regex(layerregex, options)
217     oMatches = oRegex.Matches(sContent)
218 Loop

219     'get m1 files
220     layerregex = "^\\w*[a-z]-sm1{1}\\b"
221     oRegex = New Regex(layerregex, options)
222     oMatches = oRegex.Matches(sContent)
223     For Each oMatch In oMatches
224         btImportJob.CurrentDirectory = (currpath)
225         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
226         btImportJob.LayerType(lcount) = "MSU"
227         btImportJob.LayerColor(lcount) = 9
228         btImportJob.LayerName(lcount) = Fl0Name + "m1"
229         lcount = lcount + 1

230     'Remove after debugging
231     'Console.WriteLine(oMatch.Value)

232 Next

233     'get cvl-drll files
234     layerregex = "^\\w*[a-z]-cvrly-drll"

```

```
235     oRegex = New Regex(layerregex, options)
236     oMatches = oRegex.Matches(sContent)
237     For Each oMatch In oMatches
238         btImportJob.CurrentDirectory = (currpath)
239         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
240         btImportJob.LayerType(lcount) = "MSU"
241         btImportJob.LayerColor(lcount) = 9
242         btImportJob.LayerName(lcount) = F10Name + "m1"
243         lcount = lcount + 1
244
245     'Remove after debugging
246     'Console.WriteLine(oMatch.Value)
247
248     Next
249
250     'get msl
251     layerregex = "^\\w*[a-z]-sm" & (lnum - 1) & Microsoft.VisualBasic.ChrW(
252         13) & Microsoft.VisualBasic.ChrW(10)
253     oRegex = New Regex(layerregex, options)
254     oMatches = oRegex.Matches(sContent)
255     For Each oMatch In oMatches
256         btImportJob.CurrentDirectory = (currpath)
257         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
258         btImportJob.LayerType(lcount) = "MSL"
259         btImportJob.LayerColor(lcount) = 9
260         btImportJob.LayerName(lcount) = F10Name + "m" + cpl
261         lcount = lcount + 1
262
263     'Remove after debugging
264     'Console.WriteLine(oMatch.Value)
265
266     Next
267
268     'get cvl-drl? files
269     layerregex = "^\\w*[a-z]-cvrly-drl" & (lnum - 1) & Microsoft.VisualBasic.ChrW(
270         13) & Microsoft.VisualBasic.ChrW(10)
271     oRegex = New Regex(layerregex, options)
272     oMatches = oRegex.Matches(sContent)
273     For Each oMatch In oMatches
274         btImportJob.CurrentDirectory = (currpath)
275         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
276         btImportJob.LayerType(lcount) = "MSL"
277         btImportJob.LayerColor(lcount) = 9
278         btImportJob.LayerName(lcount) = F10Name + "m" + Format(lcount)
279         lcount = lcount + 1
280
281     'Remove after debugging
282     'Console.WriteLine(oMatch.Value)
283
284     Next
285
286     'get s1
287     layerregex = "^\\w*[a-z]-ss1{1}\\k"
288     oRegex = New Regex(layerregex, options)
289     oMatches = oRegex.Matches(sContent)
290     For Each oMatch In oMatches
291         btImportJob.CurrentDirectory = (currpath)
```

```
281     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
282     btImportJob.LayerColor(lcount) = 8
283     btImportJob.LayerType(lcount) = "SKU"
284     btImportJob.LayerName(lcount) = F10Name + "s1"
285     lcount = lcount + 1

286     'Remove after debugging
287     'Console.WriteLine(oMatch.Value)

288     Next

289     'get skl
290     layerregex = "^\\w*[a-z]-ss" & (lnum - 1) & Microsoft.VisualBasic.ChrW(
291         13) & Microsoft.VisualBasic.ChrW(10)
292     oRegex = New Regex(layerregex, options)
293     oMatches = oRegex.Matches(sContent)
294     For Each oMatch In oMatches
295         btImportJob.CurrentDirectory = (currpath)
296         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
297         btImportJob.LayerColor(lcount) = 8
298         btImportJob.LayerType(lcount) = "SKL"
299         btImportJob.LayerName(lcount) = F10Name + "s" + cpl
300         lcount = lcount + 1

301     'Remove after debugging
302     'Console.WriteLine(oMatch.Value)

303     Next

304     'get map
305     layerregex = "^\\w*[a-z]?-?\\w*[a-z]-map"
306     oRegex = New Regex(layerregex, options)
307     oMatches = oRegex.Matches(sContent)
308     For Each oMatch In oMatches
309         btImportJob.CurrentDirectory = (currpath)
310         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
311         btImportJob.LayerColor(lcount) = 5
312         btImportJob.LayerType(lcount) = "BOL"
313         btImportJob.LayerName(lcount) = F10Name + "oln"
314         lcount = lcount + 1

315     'Remove after debugging
316     'Console.WriteLine(oMatch.Value)

317     Next

318     'get bv files
319     layerregex = "^\\w*[a-z]-bv[a1-9]-[0-9][0-9]?"
320     oRegex = New Regex(layerregex, options)
321     oMatches = oRegex.Matches(sContent)
322     Dim bvcnt As Integer = oMatches.Count
323     Do While bvcnt > 0
324         For Each oMatch In oMatches
325             btImportJob.CurrentDirectory = (currpath)
326             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
                ASCII, Trailing, Excellon"
            btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
```

```
327         btImportJob.LayerType(lcount) = "VID"

328     Dim message, title, defaultValue As String
329     Dim myValue As Object
330     'message = "Please name the extension"      ' Set prompt.
331     'title = "Rename Blind Via File " + oMatch.Value      ' Set title.
332     defaultValue = Right$(oMatch.Value, oMatch.Length - 10) ' Set
        default value.

333     ' Display message, title, and default value.
334     'myValue = InputBox(message, title, defaultValue)
335     myValue = defaultValue
336     btImportJob.LayerName(lcount) = F10Name + "bv" + myValue

337     'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
338     lcount = lcount + 1
339     bvcnt = bvcnt - 1

340     'Remove after debugging
341     'Console.WriteLine(oMatch.Value)

342 Next
343 Loop

344 'get lsr files
345 layerregex = "^\\w*[a-z]-lsr[1-9]-[0-9][0-9]?"
346 oRegex = New Regex(layerregex, options)
347 oMatches = oRegex.Matches(sContent)
348 Dim lsrcnt As Integer = oMatches.Count
349 Do While lsrcnt > 0
350     For Each oMatch In oMatches
351         btImportJob.CurrentDirectory = (currpath)
352         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
            ASCII, Trailing, Excellon"
353         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
354         btImportJob.LayerType(lcount) = "VID"

355         Dim message, title, defaultValue As String
356         Dim myValue As Object
357         'message = "Please name the extension"      ' Set prompt.
358         'title = "Rename Blind Via File " + oMatch.Value      ' Set title.
359         defaultValue = Right$(oMatch.Value, oMatch.Length - 9)      ' Set
            default value.

360         ' Display message, title, and default value.
361         'myValue = InputBox(message, title, defaultValue)
362         myValue = defaultValue
363         btImportJob.LayerName(lcount) = F10Name + "bv" + myValue
364         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
365         lcount = lcount + 1
366         lsrcnt = lsrcnt - 1

367         'Remove after debugging
368         'Console.WriteLine(oMatch.Value)

369     Next
370 Next Loop
```



```
371 'get csink files
372 layerregex = "^\\w*[a-z]-csink[1-9]-[0-9][0-9]?"
373 oRegex = New Regex(layerregex, options)
374 oMatches = oRegex.Matches(sContent)
375 Dim csinkcnt As Integer = oMatches.Count
376 Do While csinkcnt > 0
377     For Each oMatch In oMatches
378         btImportJob.CurrentDirectory = (currpath)
379         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
380         btImportJob.LayerType(lcount) = "VID"
381         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
382         'btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
383             ASCII, Trailing, Excellon"
384         'btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
385         'btImportJob.LayerType(lcount) = "VIA"
386         lcount = lcount + 1
387         csinkcnt = csinkcnt - 1
388     'Remove after debugging
389     'Console.WriteLine(oMatch.Value)
390 Next
391 Loop

391 'get cbore files
392 layerregex = "^\\w*[a-z]-cbore[1-9]-[0-9][0-9]?"
393 oRegex = New Regex(layerregex, options)
394 oMatches = oRegex.Matches(sContent)
395 Dim cborecnt As Integer = oMatches.Count
396 Do While cborecnt > 0
397     For Each oMatch In oMatches
398         btImportJob.CurrentDirectory = (currpath)
399         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
400             ASCII, Trailing, Excellon"
401         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
402         btImportJob.LayerType(lcount) = "VID"
403         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
404         'btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
405             ASCII, Trailing, Excellon"
406         'btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
407         'btImportJob.LayerType(lcount) = "VIA"
408         lcount = lcount + 1
409         cborecnt = cborecnt - 1
410     'Remove after debugging
411     'Console.WriteLine(oMatch.Value)
412 Next
413 Loop

412 'get dri_p file
413 layerregex = "^\\w*[a-z]-dri_p"
414 oRegex = New Regex(layerregex, options)
415 oMatches = oRegex.Matches(sContent)
416 Dim dri_pcmt As Integer = oMatches.Count
```

```
417 Do While drl_pcmt > 0
418     For Each oMatch In oMatches
419         btImportJob.CurrentDirectory = (currpath)
420         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English, ↗
            ASCII, Trailing, Excellon"
421         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
422         btImportJob.LayerColor(lcount) = 10
423         btImportJob.LayerType(lcount) = "DRI"
424         btImportJob.LayerName(lcount) = F10Name + "dri"
425         lcount = lcount + 1
426         drl_pcmt = drl_pcmt - 1

427     'Remove after debugging
428     'Console.WriteLine("drl_p_" & oMatch.Value)

429     Next
430 Loop

431     'get dri_n file
432     layerregex = "^\\w*[a-z]-drl_n"
433     oRegex = New Regex(layerregex, options)
434     oMatches = oRegex.Matches(sContent)
435     Dim drl_ncnt As Integer = oMatches.Count
436     Do While drl_ncnt > 0
437         For Each oMatch In oMatches
438             btImportJob.CurrentDirectory = (currpath)
439             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English, ↗
                ASCII, Trailing, Excellon"
440             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
441             btImportJob.LayerColor(lcount) = 10
442             btImportJob.LayerType(lcount) = "DRI"
443             btImportJob.LayerName(lcount) = F10Name + "npt"
444             ConvertNPTH()
445             lcount = lcount + 1
446             drl_ncnt = drl_ncnt - 1

447         'Remove after debugging
448         'Console.WriteLine("drl_n_" & oMatch.Value)

449         Next
450 Loop

451     'get 2nd_drill file
452     layerregex = "^\\w*[a-z]-2nd-drill"
453     oRegex = New Regex(layerregex, options)
454     oMatches = oRegex.Matches(sContent)
455     Dim secdr1 As Integer = oMatches.Count
456     Do While secdr1 > 0
457         For Each oMatch In oMatches
458             btImportJob.CurrentDirectory = (currpath)
459             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English, ↗
                ASCII, Trailing, Excellon"
460             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
461             btImportJob.LayerType(lcount) = "DRI"
462             btImportJob.LayerName(lcount) = F10Name + "npt"
463             ConvertNPTH()
464             lcount = lcount + 1
```

```
465         secdr1 = secdr1 - 1

466     'Remove after debugging
467     'Console.WriteLine("drl_n_" & oMatch.Value)

468     Next
469 Loop

470 'get epoxy files
471 layerregex = "^\\w*[a-z]-epxy[1-9]-[0-9][0-9]?"
472 oRegex = New Regex(layerregex, options)
473 oMatches = oRegex.Matches(sContent)
474 Dim epxycnt As Integer = oMatches.Count
475 Do While epxycnt > 0
476     For Each oMatch In oMatches
477         btImportJob.CurrentDirectory = (currpath)
478         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
            ASCII, Trailing, Excellon"
479         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
480         btImportJob.LayerType(lcount) = "VID"

481         Dim message, title, defaultValue As String
482         Dim myValue As Object
483         'message = "Please name the extension" ' Set prompt.
484         'title = "Rename Blind Via File " + oMatch.Value ' Set title.
485         defaultValue = Right$(oMatch.Value, oMatch.Length - 10) ' Set
            default value.

486         ' Display message, title, and default value.
487         'myValue = InputBox(message, title, defaultValue)
488         myValue = defaultValue
489         btImportJob.LayerName(lcount) = Fl0Name + "epxy" + myValue
490         'btImportJob.LayerName(lcount) = Fl0Name + Format(oMatch)
491         lcount = lcount + 1
492         epxycnt = epxycnt - 1

493     'Remove after debugging
494     'Console.WriteLine(oMatch.Value)

495     Next
496 Loop

497 'get cu files
498 layerregex = "^\\w*[a-z]-epxy[1-9]-[0-9][0-9]?"
499 oRegex = New Regex(layerregex, options)
500 oMatches = oRegex.Matches(sContent)
501 Dim cucnt As Integer = oMatches.Count
502 Do While cucnt > 0
503     For Each oMatch In oMatches
504         btImportJob.CurrentDirectory = (currpath)
505         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
            ASCII, Trailing, Excellon"
506         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
507         btImportJob.LayerType(lcount) = "VID"

508         Dim message, title, defaultValue As String
```

```

509     Dim myValue As Object
510     'message = "Please name the extension"      ' Set prompt.
511     'title = "Rename Blind Via File " + oMatch.Value      ' Set title.
512     defaultValue = Right$(oMatch.Value, oMatch.Length - 10)      ' Set
        default value.

513     ' Display message, title, and default value.
514     'myValue = InputBox(message, title, defaultValue)
515     myValue = defaultValue
516     btImportJob.LayerName(lcount) = F10Name + "cu" + myValue
517     'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
518     lcount = lcount + 1
519     cucnt = cucnt - 1

520     'Remove after debugging
521     'Console.WriteLine(oMatch.Value)

522     Next
523     Loop

524     'get slvr files
525     layerregex = "^\\w*[a-z]-slvr[1-9]-[0-9][0-9]?"
526     oRegex = New Regex(layerregex, options)
527     oMatches = oRegex.Matches(sContent)
528     Dim slvrCnt As Integer = oMatches.Count
529     Do While slvrCnt > 0
530         For Each oMatch In oMatches
531             btImportJob.CurrentDirectory = (currpath)
532             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
                ASCII, Trailing, Excellon"
533             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
534             btImportJob.LayerType(lcount) = "VID"

535             Dim message, title, defaultValue As String
536             Dim myValue As Object
537             'message = "Please name the extension"      ' Set prompt.
538             'title = "Rename Blind Via File " + oMatch.Value      ' Set title.
539             defaultValue = Right$(oMatch.Value, oMatch.Length - 10)      ' Set
                default value.

540             ' Display message, title, and default value.
541             'myValue = InputBox(message, title, defaultValue)
542             myValue = defaultValue
543             btImportJob.LayerName(lcount) = F10Name + "slvr" + myValue
544             'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
545             lcount = lcount + 1
546             slvrCnt = slvrCnt - 1

547             'Remove after debugging
548             'Console.WriteLine(oMatch.Value)

549         Next
550     Loop

551     lcount = lcount + 1

552     'tidy up and save both bin files

```

```

553     btImportJob.OnlyCurrentLayer = False
554     btImportJob.MessagesMode = "Interactive"
555     btImportJob.SaveJob(currpath & "\" & ToolNo & ".bin")
556     btImportJob.SaveJob(currpath & "\" & JobName & ".bin")
557 End Sub

558 Sub ConvertNPth()
559     Dim highlightnpth As New CAMMaster.Tool
560     With highlightnpth

561         .OnlyCurrentDCode = False
562         .OnlyCurrentLayer = True
563         .SelectGlobal()
564         Dim drilltools, dtc, dtc1, drilltcode, toolnum, Len1, ds1, dt1, ds2,
            dt2 As String

565         While .GetSelectionTools("") <> ""      '--While the selected dcodes
            is something
566             drilltools = .GetSelectionTools("")    '-----get the selected
                dcodes
567             dtc = Len(drilltools)      '--Get the length of the returned string
                of dcodes
568             dtc1 = InStr(drilltools, ",")    '--Find the location of the comma
                in the string

569             If dtc1 = 0 Then      '--If there is no comma than this is the las
                dcode
570                 drilltcode = drilltools
571             Else
572                 drilltcode = Left$(drilltools, InStr(drilltools, ",") - 1) '--I
                    there is a comma trim whatever is in front of it from the
                    left
573             End If

574             toolnum = Right$(.DCodeShape(drilltcode), Len(.DCodeShape(
                drilltcode)) - InStr(.DCodeShape(drilltcode), "=")) '--Parse th
                toolnum
575             Len1 = Len(.ToolShape(toolnum))
576             ds1 = InStrRev(.ToolShape(toolnum), ",")
577             dt1 = Len1 - ds1
578             ds2 = Left$(.ToolShape(toolnum), ds1)    '--Parse the drillsize
579             dt2 = Right$(.ToolShape(toolnum), dt1)    '--Parse the drilltype
580             .ToolShape(toolnum) = ds2 & " Non-plated"
581             .OnlyCurrentDCode = True      '--Remove the dcode from the selection
582             .CurrentDCode = drilltcode
583             .Select("Remove", -1000, -1000, 1000, 1000)
584             .OnlyCurrentDCode = False
585         End While
586     End With
587 End Sub
588 End Module

```