

```
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 = "c:\etest\temp2"
10        'currpath = "c:\etest\flex1"
11        'currpath = "C:\etest\test3"

12        'Find CPL
13        Dim cpl As String = GetSS(currpath, "")

14        'Load Files
15        GetLayers(currpath, cpl)

16        'Cleanup
17        Dim cmpfile As String = currpath & "\compare.txt"
18        Dim fi As New FileInfo(cmpfile)
19        Console.WriteLine(fi)
20        If fi.Exists Then fi.Delete()
21    End Sub

22    Public Function WriteToTempFile(ByVal Data As String) As String
23        Dim strFilename As String = Path.GetTempFileName()
24        Dim objFS As New System.IO.FileStream(strFilename, _
25        System.IO.FileMode.Append, _
26        System.IO.FileAccess.Write)
27        ' Opens stream and begins writing
28        Dim Writer As New StreamWriter(objFS)
29        Writer.BaseStream.Seek(0, SeekOrigin.End)
30        Writer.WriteLine(Data)
31        Writer.Flush()
32        ' Closes and returns temp path
33        Writer.Close()
34        Return strFilename
35    End Function

36    Function GetToolNo(ByVal currpath As String) As String
37        Dim myPath As String = currpath
38        Dim myDirectory As System.IO.DirectoryInfo
39        myDirectory = New System.IO.DirectoryInfo(myPath)
40        Dim ToolNo As String = Left(myDirectory.Name, 5)
41        Return ToolNo
42    End Function

43    Function GetSS(ByVal currpath As String, ByVal SolderSide As String) As String
44        Dim ssregex As String = "^\\w*[a-z]-[0-9]+[a-z]{1}\\k"
45        Dim options As RegexOptions = ((RegexOptions.IgnorePatternWhitespace _
46        Or RegexOptions.Multiline) _
47        Or RegexOptions.IgnoreCase)
48        Dim reg As Regex = New Regex(ssregex, options)
```

```
49     'Get directory list into sContent
50     Dim sContent
51     Dim oFile As File
52     Dim oReader As StreamReader
53     Dim oFileStream As FileStream
54     Dim oWriter As StreamWriter
55     Dim strFilename As String = WriteToTempFile("")
56     Dim file As String
57     Dim filename As String
58     Dim stripreg As String = "^.*\\\"
59     Dim stripoptions As RegexOptions = ((RegexOptions.
        IgnorePatternWhitespace _
60     Or RegexOptions.Multiline) _
61     Or RegexOptions.IgnoreCase)
62     Dim stripregex As Regex = New Regex(stripreg, stripoptions)

63     oFileStream = New FileStream(strFilename, FileMode.Create)
64     oWriter = New StreamWriter(oFileStream)

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

67         'Remove after debugging
68         'Console.WriteLine(file)

69         oWriter.WriteLine(file)
70     Next

71     oWriter.Close()
72     oFileStream.Close()
73     oReader = oFile.OpenText(strFilename)
74     sContent = oReader.ReadToEnd
75     oReader.Close()
76     oReader = Nothing
77     oFile = Nothing

78     Dim oMatches As MatchCollection
79     oMatches = reg.Matches(sContent)
80     SolderSide = Val(oMatches.Count)
81     Return SolderSide
82 End Function

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

85     Dim btImportJob As New CAMMaster.Tool
86     Dim lcount As Integer = 1
87     Dim lcolor As Integer = 1
88     Dim PlnPfx As String = "P0"
89     Dim PlnNum As Integer = 1
90     Dim ToolNo As String = GetToolNo(currpath)
91     Dim Fl0Name As String = ToolNo & "net."
92     Dim JobName As String = ToolNo & "net"

93     'Setup Expression
```

```
94 Dim lnum As Integer = 1
95 Dim layerregex As String = "^\\w*[a-z]-" & lnum & "[a-z]{1}\\b"
96 Dim options As RegexOptions = ((RegexOptions.IgnorePatternWhitespace _
97 Or RegexOptions.Multiline) _
98 Or RegexOptions.IgnoreCase)
99 Dim oRegex As Regex = New Regex(layerregex, options)
100 Dim oMatch As Match
101 Dim oMatches As MatchCollection

102 'Get directory list into sContent

103 Dim sContent
104 Dim oFile As File
105 Dim oReader As StreamReader
106 Dim oFileStream As FileStream
107 Dim oWriter As StreamWriter
108 Dim strFilename As String = WriteToTempFile("")
109 Dim file As String
110 Dim filename As String
111 Dim stripreg As String = "^.*\\\\"
112 Dim stripoptions As RegexOptions = ((RegexOptions.
    IgnorePatternWhitespace _
113 Or RegexOptions.Multiline) _
114 Or RegexOptions.IgnoreCase)
115 Dim stripregex As Regex = New Regex(stripreg, stripoptions)

116 oFileStream = New FileStream(strFilename, FileMode.Create)
117 oWriter = New StreamWriter(oFileStream)

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

120     'Remove after debugging
121     'Console.WriteLine(file)

122     oWriter.WriteLine(file)
123 Next

124 oWriter.Close()
125 oFileStream.Close()
126 oReader = oFile.OpenText(strFilename)
127 sContent = oReader.ReadToEnd
128 oReader.Close()
129 oReader = Nothing
130 oFile = Nothing

131 'get gerber layers
132 'kill screen updates in CAMMAster
133 btImportJob.MessagesMode = "Background"
134 oMatches = oRegex.Matches(sContent)
135 If oMatches.Count = 0 And lnum = 1 Then
136     Console.WriteLine("No Layer 1 Found")
137     Console.WriteLine("Press ENTER to QUIT")
138     Console.ReadLine()
139 End
140 End If
```

```

141 Do While oMatches.Count = 1
142 For Each oMatch In oMatches
143     btImportJob.CurrentDirectory = (currpath)
144     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)

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

```

```

    + "d"
189     'btImportJob.LayerExtension(lcount) = ".g" + lcount + "d"
190   -Else
191     btImportJob.LayerNegative(lcount) = False
192     btImportJob.LayerType(lcount) = "SIG"
193     btImportJob.LayerName(lcount) = F10Name + Format(lcount)
194     'btImportJob.LayerExtension(lcount) = "." + lcount
195     btImportJob.LayerColor(lcount) = lcolor
196   -End If
197     lcount = lcount + 1
198   -End If

199   'Remove after debugging
200   'Console.WriteLine(oMatch.Value)

201   -Next
202     lnum = lnum + 1
203     lcolor = lcolor + 1
204     layerregex = "^\\w*[a-z]-" & lnum & "[a-z]{1}\\b"
205     oRegex = New Regex(layerregex, options)
206     oMatches = oRegex.Matches(sContent)
207   -Loop

208   'get m1 files
209   layerregex = "^\\w*[a-z]-sm1{1}\\b"
210   oRegex = New Regex(layerregex, options)
211   oMatches = oRegex.Matches(sContent)
212   -For Each oMatch In oMatches
213     btImportJob.CurrentDirectory = (currpath)
214     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
215     btImportJob.LayerType(lcount) = "MSU"
216     btImportJob.LayerColor(lcount) = 9
217     btImportJob.LayerName(lcount) = F10Name + "m1"
218     lcount = lcount + 1

219   'Remove after debugging
220   'Console.WriteLine(oMatch.Value)

221   -Next

222   'get cvl-drll files
223   layerregex = "^\\w*[a-z]-cvrly-drll"
224   oRegex = New Regex(layerregex, options)
225   oMatches = oRegex.Matches(sContent)
226   -For Each oMatch In oMatches
227     btImportJob.CurrentDirectory = (currpath)
228     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
229     btImportJob.LayerType(lcount) = "MSU"
230     btImportJob.LayerColor(lcount) = 9
231     btImportJob.LayerName(lcount) = F10Name + "m1"
232     lcount = lcount + 1

233   'Remove after debugging
234   'Console.WriteLine(oMatch.Value)
```

```
235     Next

236     'get msl
237     layerregex = "^\\w*[a-z]-sm" & (lnum - 1) & Microsoft.VisualBasic.ChrW(
13) & Microsoft.VisualBasic.ChrW(10)
238     oRegex = New Regex(layerregex, options)
239     oMatches = oRegex.Matches(sContent)
240     For Each oMatch In oMatches
241         btImportJob.CurrentDirectory = (currpath)
242         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
243         btImportJob.LayerType(lcount) = "MSL"
244         btImportJob.LayerColor(lcount) = 9
245         btImportJob.LayerName(lcount) = F10Name + "m" + cpl
246         lcount = lcount + 1

247     'Remove after debugging
248     'Console.WriteLine(oMatch.Value)

249     Next

250     'get cvl-drl? files
251     layerregex = "^\\w*[a-z]-cvrly-drl" & (lnum - 1) & Microsoft.VisualBasic.ChrW(13) & Microsoft.VisualBasic.ChrW(10)
252     oRegex = New Regex(layerregex, options)
253     oMatches = oRegex.Matches(sContent)
254     For Each oMatch In oMatches
255         btImportJob.CurrentDirectory = (currpath)
256         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
257         btImportJob.LayerType(lcount) = "MSL"
258         btImportJob.LayerColor(lcount) = 9
259         btImportJob.LayerName(lcount) = F10Name + "m" + Format(lcount)
260         lcount = lcount + 1

261     'Remove after debugging
262     'Console.WriteLine(oMatch.Value)

263     Next

264     'get sl
265     layerregex = "^\\w*[a-z]-ss1{1}\\k"
266     oRegex = New Regex(layerregex, options)
267     oMatches = oRegex.Matches(sContent)
268     For Each oMatch In oMatches
269         btImportJob.CurrentDirectory = (currpath)
270         btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
271         btImportJob.LayerColor(lcount) = 8
272         btImportJob.LayerType(lcount) = "SKU"
273         btImportJob.LayerName(lcount) = F10Name + "s1"
274         lcount = lcount + 1

275     'Remove after debugging
276     'Console.WriteLine(oMatch.Value)

277     Next

278     'get skl
```

```
279 layerregex = "^\\w*[a-z]-ss" & (lnum - 1) & Microsoft.VisualBasic.ChrW(
    13) & Microsoft.VisualBasic.ChrW(10)
280 oRegex = New Regex(layerregex, options)
281 oMatches = oRegex.Matches(sContent)
282 For Each oMatch In oMatches
283     btImportJob.CurrentDirectory = (currpath)
284     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
285     btImportJob.LayerColor(lcount) = 8
286     btImportJob.LayerType(lcount) = "SKL"
287     btImportJob.LayerName(lcount) = F10Name + "s" + cpl
288     lcount = lcount + 1

289 'Remove after debugging
290 'Console.WriteLine(oMatch.Value)

291 Next

292 'get map
293 layerregex = "^\\w*[a-z]?-?\\w*[a-z]-map"
294 oRegex = New Regex(layerregex, options)
295 oMatches = oRegex.Matches(sContent)
296 For Each oMatch In oMatches
297     btImportJob.CurrentDirectory = (currpath)
298     btImportJob.ImportX(oMatch.Value, "GERBER", lcount, 0, 0)
299     btImportJob.LayerColor(lcount) = 5
300     btImportJob.LayerType(lcount) = "BOL"
301     btImportJob.LayerName(lcount) = F10Name + "oln"
302     lcount = lcount + 1

303 'Remove after debugging
304 'Console.WriteLine(oMatch.Value)

305 Next

306 'get bv files
307 layerregex = "^\\w*[a-z]-bvvia[1-9]-[0-9][0-9]?"
308 oRegex = New Regex(layerregex, options)
309 oMatches = oRegex.Matches(sContent)
310 Dim bvcnt As Integer = oMatches.Count
311 Do While bvcnt > 0
312     For Each oMatch In oMatches
313         btImportJob.CurrentDirectory = (currpath)
314         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
            ASCII, Trailing, Excellon"
315         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
316         btImportJob.LayerType(lcount) = "VID"

317         Dim message, title, defaultValue As String
318         Dim myValue As Object
319         'message = "Please name the extension" ' Set prompt.
320         'title = "Rename Blind Via File " + oMatch.Value ' Set title.
321         defaultValue = Right$(oMatch.Value, oMatch.Length - 10) ' Set
            default value.

322         ' Display message, title, and default value.
323         'myValue = InputBox(message, title, defaultValue)
324         myValue = defaultValue
```

```
325         btImportJob.LayerName(lcount) = F10Name + "bv" + myValue
326
327         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
328         lcount = lcount + 1
329         bvcnt = bvcnt - 1
330
331     'Remove after debugging
332     'Console.WriteLine(oMatch.Value)
333
334     Next
335 Loop
336
337 'get lsr files
338 layerregex = "^\\w*[a-z]-lsr[1-9]-[0-9][0-9]?"
339 oRegex = New Regex(layerregex, options)
340 oMatches = oRegex.Matches(sContent)
341 Dim lsrcnt As Integer = oMatches.Count
342 Do While lsrcnt > 0
343     For Each oMatch In oMatches
344         btImportJob.CurrentDirectory = (currpath)
345         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
346             ASCII, Trailing, Excellon"
347         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
348         btImportJob.LayerType(lcount) = "VID"
349
350         Dim message, title, defaultValue As String
351         Dim myValue As Object
352         'message = "Please name the extension" ' Set prompt.
353         'title = "Rename Blind Via File " + oMatch.Value ' Set title.
354         'defaultValue = Right$(oMatch.Value, oMatch.Length - 9) ' Set
355         'default value.
356
357         ' Display message, title, and default value.
358         'myValue = InputBox(message, title, defaultValue)
359         myValue = defaultValue
360         btImportJob.LayerName(lcount) = F10Name + "bv" + myValue
361         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
362         lcount = lcount + 1
363         lsrcnt = lsrcnt - 1
364
365     'Remove after debugging
366     'Console.WriteLine(oMatch.Value)
367
368     Next
369 Loop
370
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)
```



```
371         'btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
          ASCII, Trailing, Excellon"
372         'btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
373         'btImportJob.LayerType(lcount) = "VIA"
374         lcount = lcount + 1
375         csinkcnt = csinkcnt - 1

376     'Remove after debugging
377     'Console.WriteLine(oMatch.Value)

378     Next
379 Loop

380     'get cbore files
381     layerregex = "^\\w*[a-z]-cbore[1-9]-[0-9][0-9]?"
382     oRegex = New Regex(layerregex, options)
383     oMatches = oRegex.Matches(sContent)
384     Dim cborecnt As Integer = oMatches.Count
385     Do While cborecnt > 0
386     For Each oMatch In oMatches
387         btImportJob.CurrentDirectory = (currpath)
388         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
          ASCII, Trailing, Excellon"
389         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
390         btImportJob.LayerType(lcount) = "VID"
391         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
392         'btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
          ASCII, Trailing, Excellon"
393         'btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
394         'btImportJob.LayerType(lcount) = "VIA"
395         lcount = lcount + 1
396         cborecnt = cborecnt - 1

397     'Remove after debugging
398     'Console.WriteLine(oMatch.Value)

399     Next
400 Loop

401     'get dri_p file
402     layerregex = "^\\w*[a-z]-dri_p"
403     oRegex = New Regex(layerregex, options)
404     oMatches = oRegex.Matches(sContent)
405     Dim dri_pcmt As Integer = oMatches.Count
406     Do While dri_pcmt > 0
407     For Each oMatch In oMatches
408         btImportJob.CurrentDirectory = (currpath)
409         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
          ASCII, Trailing, Excellon"
410         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
411         btImportJob.LayerColor(lcount) = 10
412         btImportJob.LayerType(lcount) = "DRI"
413         btImportJob.LayerName(lcount) = F10Name + "dri"
414         lcount = lcount + 1
415         dri_pcmt = dri_pcmt - 1
```

```
416     'Remove after debugging
417     'Console.WriteLine("drl_p_" & oMatch.Value)

418     Next
419 Loop

420     'get dri_n file
421     layerregex = "^\\w*[a-z]-drl_n"
422     oRegex = New Regex(layerregex, options)
423     oMatches = oRegex.Matches(sContent)
424     Dim drl_ncnt As Integer = oMatches.Count
425     Do While drl_ncnt > 0
426         For Each oMatch In oMatches
427             btImportJob.CurrentDirectory = (currpath)
428             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
                ASCII, Trailing, Excellon"
429             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
430             btImportJob.LayerColor(lcount) = 10
431             btImportJob.LayerType(lcount) = "DRI"
432             btImportJob.LayerName(lcount) = F10Name + "npt"
433             ConvertNPTH()
434             lcount = lcount + 1
435             drl_ncnt = drl_ncnt - 1

436         'Remove after debugging
437         'Console.WriteLine("drl_n_" & oMatch.Value)

438     Next
439 Loop

440     'get 2nd_drill file
441     layerregex = "^\\w*[a-z]-2nd-drl"
442     oRegex = New Regex(layerregex, options)
443     oMatches = oRegex.Matches(sContent)
444     Dim secdr1 As Integer = oMatches.Count
445     Do While secdr1 > 0
446         For Each oMatch In oMatches
447             btImportJob.CurrentDirectory = (currpath)
448             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
                ASCII, Trailing, Excellon"
449             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
450             btImportJob.LayerType(lcount) = "DRI"
451             btImportJob.LayerName(lcount) = F10Name + "npt"
452             ConvertNPTH()
453             lcount = lcount + 1
454             secdr1 = secdr1 - 1

455         'Remove after debugging
456         'Console.WriteLine("drl_n_" & oMatch.Value)

457     Next
458 Loop

459     'get epoxy files
460     layerregex = "^\\w*[a-z]-epxy[1-9]-[0-9][0-9]?"
461     oRegex = New Regex(layerregex, options)
```

```
462 oMatches = oRegex.Matches(sContent)
463 Dim epxycnt As Integer = oMatches.Count
464 Do While epxycnt > 0
465     For Each oMatch In oMatches
466         btImportJob.CurrentDirectory = (currpath)
467         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
ASCII, Trailing, Excellon"
468         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
469         btImportJob.LayerType(lcount) = "VID"

470         Dim message, title, defaultValue As String
471         Dim myValue As Object
472         'message = "Please name the extension" ' Set prompt.
473         'title = "Rename Blind Via File " + oMatch.Value ' Set title.
474         defaultValue = Right$(oMatch.Value, oMatch.Length - 10) ' Set
default value.

475         ' Display message, title, and default value.
476         'myValue = InputBox(message, title, defaultValue)
477         myValue = defaultValue
478         btImportJob.LayerName(lcount) = F10Name + "epxy" + myValue
479         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
480         lcount = lcount + 1
481         epxycnt = epxycnt - 1

482         'Remove after debugging
483         'Console.WriteLine(oMatch.Value)

484     Next
485 Loop

486 'get cu files
487 layerregex = "^\\w*[a-z]-epxy[1-9]-[0-9][0-9]?"
488 oRegex = New Regex(layerregex, options)
489 oMatches = oRegex.Matches(sContent)
490 Dim cucnt As Integer = oMatches.Count
491 Do While cucnt > 0
492     For Each oMatch In oMatches
493         btImportJob.CurrentDirectory = (currpath)
494         btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
ASCII, Trailing, Excellon"
495         btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
496         btImportJob.LayerType(lcount) = "VID"

497         Dim message, title, defaultValue As String
498         Dim myValue As Object
499         'message = "Please name the extension" ' Set prompt.
500         'title = "Rename Blind Via File " + oMatch.Value ' Set title.
501         defaultValue = Right$(oMatch.Value, oMatch.Length - 10) ' Set
default value.

502         ' Display message, title, and default value.
503         'myValue = InputBox(message, title, defaultValue)
504         myValue = defaultValue
505         btImportJob.LayerName(lcount) = F10Name + "cu" + myValue
506         'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
507         lcount = lcount + 1
```

```
508         cucnt = cucnt - 1

509     'Remove after debugging
510     'Console.WriteLine(oMatch.Value)

511     Next
512 Loop

513     'get slvr files
514     layerregex = "^\\w*[a-z]-slvr[1-9]-[0-9][0-9]?"
515     oRegex = New Regex(layerregex, options)
516     oMatches = oRegex.Matches(sContent)
517     Dim slvrCnt As Integer = oMatches.Count
518     Do While slvrCnt > 0
519         For Each oMatch In oMatches
520             btImportJob.CurrentDirectory = (currpath)
521             btImportJob.DrillDataFormat = "Left=2, Right=4, Absolute, English,
                    ASCII, Trailing, Excellon"
522             btImportJob.ImportX(oMatch.Value, "Drill", lcount, 0, 0)
523             btImportJob.LayerType(lcount) = "VID"

524             Dim message, title, defaultValue As String
525             Dim myValue As Object
526             'message = "Please name the extension"      ' Set prompt.
527             'title = "Rename Blind Via File " + oMatch.Value      ' Set title.
528             defaultValue = Right$(oMatch.Value, oMatch.Length - 10)      ' Set
                    default value.

529             ' Display message, title, and default value.
530             'myValue = InputBox(message, title, defaultValue)
531             myValue = defaultValue
532             btImportJob.LayerName(lcount) = F10Name + "slvr" + myValue
533             'btImportJob.LayerName(lcount) = F10Name + Format(oMatch)
534             lcount = lcount + 1
535             slvrCnt = slvrCnt - 1

536             'Remove after debugging
537             'Console.WriteLine(oMatch.Value)

538         Next
539     Loop

540     lcount = lcount + 1

541     'tidy up and save both bin files
542     btImportJob.OnlyCurrentLayer = False
543     btImportJob.MessagesMode = "Interactive"
544     btImportJob.SaveJob(currpath & "\" & ToolNo & ".bin")
545     btImportJob.SaveJob(currpath & "\" & JobName & ".bin")
546 End Sub

547 Sub ConvertNPth()
548     Dim highlightNpth As New CAMMaster.Tool
549     With highlightNpth

550         .OnlyCurrentDCode = False
551         .OnlyCurrentLayer = True
```

```
552 .SelectGlobal()  
553 Dim drilltools, dtc, dtc1, drilltcode, toolnum, Len1, ds1, dt1, ds2, dt2 As String  
  
554 While .GetSelectionTools("") <> "" '--While the selected dcodes is something  
555 drilltools = .GetSelectionTools("") '-----get the selected dcodes  
556 dtc = Len(drilltools) '--Get the length of the returned string of dcodes  
557 dtc1 = InStr(drilltools, ",") '--Find the location of the comma in the string  
  
558 If dtc1 = 0 Then '--If there is no comma than this is the last dcode  
559 drilltcode = drilltools  
560 Else  
561 drilltcode = Left$(drilltools, InStr(drilltools, ",") - 1) '--If there is a comma trim whatever is in front of it from the left  
562 End If  
  
563 toolnum = Right$(.DCodeShape(drilltcode), Len(.DCodeShape(drilltcode)) - InStr(.DCodeShape(drilltcode), "=")) '--Parse the toolnum  
564 Len1 = Len(.ToolShape(toolnum))  
565 ds1 = InStrRev(.ToolShape(toolnum), ",")  
566 dt1 = Len1 - ds1  
567 ds2 = Left$(.ToolShape(toolnum), ds1) '--Parse the drillsize  
568 dt2 = Right$(.ToolShape(toolnum), dt1) '--Parse the drilltype  
569 .ToolShape(toolnum) = ds2 & " Non-plated"  
570 .OnlyCurrentDCode = True '--Remove the dcode from the selection  
571 .CurrentDCode = drilltcode  
572 .Select("Remove", -1000, -1000, 1000, 1000)  
573 .OnlyCurrentDCode = False  
574 End While  
575 End With  
576 End Sub  
577 End Module
```

```
1 Imports System
2 Imports System.Reflection
3 Imports System.Runtime.InteropServices

4 ' General Information about an assembly is controlled through the following
5 ' set of attributes. Change these attribute values to modify the information
6 ' associated with an assembly.

7 ' Review the values of the assembly attributes

8 <Assembly: AssemblyTitle("netprep")>
9 <Assembly: AssemblyDescription("Load data into CAMMaster")>
10 <Assembly: AssemblyCompany("Speedy Circuits, Inc.")>
11 <Assembly: AssemblyProduct("netprep")>
12 <Assembly: AssemblyCopyright("2005-2006 PJC Technologies, Inc.")>
13 <Assembly: AssemblyTrademark("")>
14 <Assembly: CLSCompliant(True)>

15 'The following GUID is for the ID of the typelib if this project is exposed
    to COM
16 <Assembly: Guid("B2F9449E-9664-4C81-8E67-B4D110FE6FD8")>

17 ' Version information for an assembly consists of the following four
    values:
18 '
19 '         Major Version
20 '         Minor Version
21 '         Build Number
22 '         Revision
23 '
24 ' You can specify all the values or you can default the Build and Revision
    Numbers
25 ' by using the '*' as shown below:

26 <Assembly: AssemblyVersion("1.0.*")>
```