

Sub Process_data()

'written by Paul Dunn (dunnsept @ gmail dot com)
'Feb 2010
'Macro to process vmware vscsiStats data
'written in and tested in Excel 2003
'it will expect your data to be in column A and the histogram BINS to be in column B
'it will create a number of chart-sheets
'charts will be created on individual tabs. If you run
'vscsistats -p all -w WID and you have lots of drives this can make for a large
'and unwieldy spreadsheet. If you have lots of drives, I recommend that you
'process drives seperately
,

'Modified by Matt Kelliher (mattkelliher at gmail dot com)
'Mar 2010
'To make things easier to read, create a simple HTML page
'and export all the charts to GIF (storing them in subfolder)
'and display them on the HTML page (resized)
,

'Modified by Paul Dunn (dunnsept @ gmail dot com)
'changed file name to include worldgroupid, added worldgroupid
'to page title and heading
'multiple runs shouldn't overwrite as long as worldgroupid is unique
'minor tweaks

Dim count As Integer
Dim start As Integer
Dim cur As String
cur = ActiveSheet.Name

Sheets(cur).Select
start = 7
count = 7
Range("A" & start).Select

'Create a simple HTML page and images folder to display the results.
'NOTE: this will create both the index.html page and a subfolder called images
'wherever your current Excel worksheet is saved/opened. If a folder called images
'already exists then we'll just use it and overwrite any image files with the same
'name.
,

Dim fso
Dim fileobj
Dim imgfolder As String
Dim file As String
imgfolder = ActiveWorkbook.Path & "\images" 'the path from the current working directory plus the new images folder
Set fso = CreateObject("Scripting.FileSystemObject")
If Not fso.FolderExists(imgfolder) Then 'if we don't see a folder called images, create it
fso.CreateFolder (imgfolder)
End If
file = ActiveWorkbook.Path & "\index" & Range("c1").Value & ".html" 'the file is saved wherever we're at with the current worksheet
Dim ts
'ts = fso.CreateTextFile(file, True)
fso.CreateTextFile file 'create the text file
Set fileobj = fso.GetFile(file)
Set ts = fileobj.OpenAsTextStream(2, -2) 'open the text file for writing
'Write a header to the HTML file using XHTML 1.0 spec --really not necessary, but might as well *try* to play nice
ts.WriteLine ("<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" ""http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"">")
ts.WriteLine ("<html xmlns=""http://www.w3.org/1999/xhtml"">")
ts.WriteLine ("<head>")
ts.WriteLine ("<meta http-equiv=""Content-Type"" content=""text/html; charset=utf-8"" />")
ts.WriteLine ("<title>VM vscsiStats for WorldGroup ID " & Range("c1").Value & "</title>")
ts.WriteLine ("</head>")
ts.WriteLine ("<body>")
ts.WriteLine ("<h1>VM vscsiStats for WorldGroup ID " & Range("c1").Value & "</h1>")
ts.WriteLine ("<p>Click a thumbnail to view the full image in a new browser window.</p>")
'Feel free to insert any other flowery HTML code as you see fit
,

```
Do Until IsEmpty(ActiveCell)
```

```
' Set Do loop to stop on empty cell to count how many data rows we have
```

```
Do Until ((InStr(1, ActiveCell.Value, "Histogram", vbTextCompare) Or (IsEmpty(ActiveCell)))
```

```
count = count + 1
```

```
' Step down 1 row from present location.
```

```
ActiveCell.Offset(1, 0).Select
```

```
Loop
```

```
'so we have start and count. start is top of list, count is bottom of list
```

```
Range("G" & start).Value = "START OF HISTOGRAM DATA"
```

```
Range("G" & count).Value = "END OF HISTOGRAM DATA"
```

```
'we have start of data, end of data, create the chart
```

```
Dim charting As String 'we'll store the name of the chart image when it's returned from the create_chart function
```

```
charting = create_chart(start, count, cur, imgfolder)
```

```
'write out more HTML code for our chart image to display thumbnails and hyperlinks to the full version
```

```
ts.Write ("<a target=""_blank"" href=""images\" & charting & "">")
```

```
ts.Write ("<img src=""images\" & charting & "" width=""25%"" height=""25%"" />")
```

```
ts.Write ("</a>&nbsp;")
```

```
ts.WriteLine ("")
```

```
'reset and start looking again
```

```
start = count + 6
```

```
count = count + 6
```

```
Sheets(cur).Select
```

```
Range("a" & start).Select
```

```
Loop
```

```
'Write the final html code and close up the index.html file
```

```
ts.WriteLine ("</body></html>")
```

```
ts.Close
```

```
End Sub
```

```
Function create_chart(st As Integer, en As Integer, Sheet1 As String, imgfolder As String) As String
```

```
'creates a chart :-)
```

```
'Feb 2010 Paul Dunn
```

```
,
```

```
'Modified Mar 2010 Matt Kelliher
```

```
,
```

```
'again by Paul
```

```
Dim chartname As String
```

```
Dim chartfile As String
```

```
,
```

```
'read the histogram type, then figure out if it's overall, read or write
```

```
'create a new sheet to hold it.
```

```
,
```

```
'IO Length Charts
```

```
If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "lengths", vbTextCompare) Then
```

```
    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Read", vbTextCompare) Then
```

```
        chartname = "Read IOLength " & Sheets(Sheet1).Range("e" & st - 6).Value
```

```
    ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Write", vbTextCompare) Then
```

```
        chartname = "Write IOLength " & Sheets(Sheet1).Range("e" & st - 6).Value
```

```
    Else
```

```
        chartname = "IOLength " & Sheets(Sheet1).Range("e" & st - 6).Value
```

```
    End If
```

```
,
```

```
'seek distance
```

```
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "LBNS", vbTextCompare) Then
```

```
    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Read", vbTextCompare) Then
```

```
        chartname = "Read SeekDistance " & Sheets(Sheet1).Range("e" & st - 6).Value
```

```

ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Write", vbTextCompare) Then
    chartname = "Write SeekDistance " & Sheets(Sheet1).Range("e" & st - 6).Value

ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "closest", vbTextCompare) Then
    chartname = "Closest SeekDistance " & Sheets(Sheet1).Range("e" & st - 6).Value

Else
    chartname = "SeekDistance " & Sheets(Sheet1).Range("e" & st - 6).Value

End If
,
'interarrival latency charts
,
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "interarrival", vbTextCompare) Then
    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Read", vbTextCompare) Then
        chartname = "Interarrival Read Latency " & Sheets(Sheet1).Range("e" & st - 6).Value

    ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Write", vbTextCompare) Then
        chartname = "Interarrival Write " & Sheets(Sheet1).Range("e" & st - 6).Value

    Else
        chartname = "Interarrival Latency " & Sheets(Sheet1).Range("e" & st - 6).Value

    End If

'Latency charts
ElseIf (InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "latency", vbTextCompare) And Not ((InStr(1, Sheets(Sheet1).Range("A" & st -
6).Value, "interarrival", vbTextCompare)))) Then

    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Read", vbTextCompare) Then
        chartname = "Read Latency " & Sheets(Sheet1).Range("e" & st - 6).Value

    ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Write", vbTextCompare) Then
        chartname = "Write Latency " & Sheets(Sheet1).Range("e" & st - 6).Value

    Else
        chartname = "Latency " & Sheets(Sheet1).Range("e" & st - 6).Value

    End If
,
,
'outstanding IO charts
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "outstanding", vbTextCompare) Then
    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Read", vbTextCompare) Then
        chartname = "Outstanding Read IOs " & Sheets(Sheet1).Range("e" & st - 6).Value

    ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "Write", vbTextCompare) Then
        chartname = "Outstanding Write IOs " & Sheets(Sheet1).Range("e" & st - 6).Value

    Else
        chartname = "Outstanding IOs " & Sheets(Sheet1).Range("e" & st - 6).Value

    End If
End If

Charts.Add
ActiveChart.ChartType = xlColumnClustered
ActiveChart.SetSourceData Source:=Sheets(Sheet1).Range("A" & st & ":A" & en - 1), PlotBy:=_
xlColumns
ActiveChart.SeriesCollection(1).XValues = "=" & Sheet1 & "!R" & st & "C2:R" & en & "C2"
ActiveChart.Location Where:=xlLocationAsNewSheet, Name:=chartname
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = Sheets(Sheet1).Range("A" & st - 6).Value & " Volume: " & Sheets(Sheet1).Range("e" & st - 6).Value

    .Axes(xlCategory, xlPrimary).HasTitle = True

    If InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "lengths", vbTextCompare) Then
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Bytes"
    End If
End With

```

```
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "LBNs", vbTextCompare) Then
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "LBN"
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "latency", vbTextCompare) Then
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "uSec"
ElseIf InStr(1, Sheets(Sheet1).Range("A" & st - 6).Value, "outstanding", vbTextCompare) Then
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "# of IOs"
End If
```

```
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Freq"
End With
ActiveChart.Legend.Select
Selection.Delete
```

```
'Some code to export the chart as a PNG file
chartfile = imgfolder & "\" & chartname & ".PNG"
ActiveChart.Export chartfile, "PNG", False
create_chart = chartname & ".PNG"
```

End Function