**LogParser, PowerShell and Quick and dirty parsing of IIS files**

For a local enterprise IIS system you do not have to resort to Google analytics or other beasts that interprets IIs logs. After all, users are identified through ActiveDirectory, does not matter from what city do they come, and so on. But it will help to have some details of wjhat happened on the system this day( or the day before , if you want to send an email about the previous day totals)

So [logparser](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en) to help  – he knows already to read IIS logs with  -i:IISW3C

So I have come up with the syntax  :

**C:\**LogParser -e:10 -i:IISW3C “SELECT cs-uri-stem as url, DIV(SUM(time-taken),1000) as Seconds, Count(time-taken) as Requests, DIV(Seconds ,Requests) as TimeExecuting   INTO ***C:\newfile*** FROM   ***C:\Windows\System32\LogFiles\W3SVC1\ex100909.log*** GROUP BY cs-uri-stem Having SUM(time-taken)>0 and Seconds>0 order by Seconds   desc” -o:TPL -tpl:%2\iistime.tpl

Basically , this will do this report about statuses of URL requested  :

|  |  |
| --- | --- |
| **Status** | **Requests** |
| 200 | 1541 |
| 302 | 89 |
| 401 | 11 |
| 403 | 61 |

The problem is that ***C:\newfile*** and  ***C:\Windows\System32\LogFiles\W3SVC1\ex100909.log*** are hard-coded – we need to modify every time… So PowerShell to the rescue (Ok, I could do a C# Console program – but

1. it is more fun this way – fun meaning I want to learn something new

2. the script could be modified easily

)

So the same command is written this way with arguments , in order to can be executed each time :

|  |  |
| --- | --- |
| 1 | %2\LogParser -e:10 -i:IISW3C "SELECT cs-uri-stem as url, DIV(SUM(time-taken),1000) as Seconds, Count(time-taken) as Requests, DIV(Seconds ,Requests) as TimeExecuting   INTO %2\%4 FROM   %5\\*%1  GROUP BY cs-uri-stem Having SUM(time-taken)&gt;0 and Seconds&gt;0 order by Seconds   desc" -o:TPL -tpl:%2\iistime.tpl |

But who will give arguments ( such as the system date ) ?Now powershell  to the rescue :

|  |  |  |
| --- | --- | --- |
| 01 | $namepc = (gc env:computername) | |
| 02 | $a = get-date |

|  |  |
| --- | --- |
| 03 | $a = (get-date).AddDays(-1) |
| 04 | $allpath= Split-Path -Parent $MyInvocation.MyCommand.Path; | |

|  |  |  |
| --- | --- | --- |
| 05 | $logfolders = $env:WINDIR +"\system32\Logfiles\W3SVC\*" | |
| 06 | foreach($logfolder in  Get-ChildItem $logfolders) |

|  |  |
| --- | --- |
| 07 | { |
| 08 | $logfiles= $logfolder.FullName | |

|  |  |
| --- | --- |
| 09 | Write-Host "parsing"  $logfiles |
| 10 | $log =  $a.ToString("yyMMdd") + ".log" | |

|  |  |  |
| --- | --- | --- |
| 11 | $process = [Diagnostics.Process]::Start($allpath + "\iis.bat" , $log + " "+ $allpath + " "+ $log + ".html" + " TIME" +$log + ".html" + " " + $logfiles) | |
| 12 | $process.WaitForExit() |

|  |  |  |
| --- | --- | --- |
| 13 | $content = "&lt;h1&gt;IIS REPORT " + $namepc  + "&lt;/H1&gt;" | |
| 14 | $content += (get-content ($allpath  + "\" + $log + ".html")) |

|  |  |  |
| --- | --- | --- |
| 15 | $content += (get-content ($allpath  + "\TIME" + $log + ".html")) | |
| 16 |  |

|  |  |  |
| --- | --- | --- |
| 17 | $SmtpClient = new-object system.net.mail.smtpClient | |
| 18 | $SmtpServer = "your server" |

|  |  |  |
| --- | --- | --- |
| 19 | $SmtpClient.host = $SmtpServer | |
| 20 |  |

|  |  |  |
| --- | --- | --- |
| 21 | $mm = new-Object System.Net.Mail.MailMessage(“<a href="mailto:from@yourcompany.com">from@yourcompany.com</a>”,"<a href="mailto:to@yourcompany.com">to@yourcompany.com</a>") | |
| 22 | $mm.Subject = "Report IIS " + $namepc |

|  |  |
| --- | --- |
| 23 | $mm.Body = $content |
| 24 | $mm.Body=$mm.Body.Replace("&lt;cmp&gt;",$namepc  ) | |

|  |  |
| --- | --- |
| 25 | $mm.IsBodyHtml = 1 |
| 26 | $SmtpClient.Send($mm) | |

|  |  |
| --- | --- |
| 27 | } |

Explanation of code :  
line 1: I take the computer name to put in the report  
line 2 : take the date ( if you want the current date , just comment the line 3)  
line 5 : I go to usual path to logfiles (did I say quick and dirty ?)  
line 6 : get all W3SVC folders and iterate to send report  
line 11 : launching the bat (that contains logparser command ) in order to parse  arguments  
line 12 :waiting for the process to exit – in order to can send files.  
line 13 to 15: get the output  
line 17 to 27 : send the output by email

Homework :  
1. Execute script on a system and modify if it does not work  
2. Clean up the temporary files after sending email  
3. Instead of sending an email, write into a database with the current date.

You can execute sc.bat at regular times ( such as 1:00 AM)

Here is the zip file with sources [logiisparser](http://msprogrammer.serviciipeweb.ro/wp-content/uploads/2010/09/logiisparser.zip)

LogParser download : <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en>

Powershell scripts : [http://gallery.technet.microsoft.com/ScriptCenter/en-us/site/search?f[1].Type=SearchText&f[1].Value=internet&f[0].Value=applications&f[0].Type=RootCategory&f[0].Text=Applications&x=0&y=0](http://gallery.technet.microsoft.com/ScriptCenter/en-us/site/search?f%5b1%5d.Type=SearchText&f%5b1%5d.Value=internet&f%5b0%5d.Value=applications&f%5b0%5d.Type=RootCategory&f%5b0%5d.Text=Applications&x=0&y=0)