Clean-IISLogFiles.ps1

```
   <#
.SYNOPSIS
    Delete IIS Log Files that exist on the target if they are over a specific age.

.DESCRIPTION
    A default value of 30 days is configured in the script.  You can also create
    a JSON configuration file named 'LogRetentionConfiguration.json' and place it
    in the LogFiles folder.  The thresholds specified in that JSON file will then
    be utilised instead; if no value is provided in the file for a specific website
    folder then the default will be applied.

    Sample Json format:

        {"LogRetentionTimes":[
            {"W3SVC1":{"LogRetentionTime":"20"}},
            {"W3SVC2":{"LogRetentionTime":"10"}}
        ]}

.PARAMETER Format
    Default retention period should be specified as a numeric.  If not specified then
    a default of 30 is utilised.
.EXAMPLE
.NOTES
.LINK
#>
Param(
    [int]$DefaultLogRetention = 30
)

# display status
Write-Host "Starting Cleanup of IIS Log Files"
Write-Host ""
Write-Host " Checking if the WebAdministration PowerShell Module is present: " -NoNewLine

# if the machine has the web administration module then...
if ((Get-Module -ListAvailable WebAdministration) -ne $null)
{
       # display status
       Write-Host "OK"

       # import the web administration module
       Import-Module WebAdministration

       # get the list of websites
       $webSites = Get-Website

       # if there are some websites
       if ($webSites -ne $null)
       {
        # if it's only a single website then...
              if ($webSites -isnot [Array])
        {
            # get the path to the first log file
                 $logPath =
[System.Environment]::ExpandEnvironmentVariables($WebSites.LogFile.Directory)
        }
        else
        {
            # get the path to the first log file (using the index of the array)
                 $logPath =
[System.Environment]::ExpandEnvironmentVariables($WebSites[0].LogFile.Directory)
        }

              # append another slash and the configuration file we might expect to see
              $logPath += "\LogRetentionConfiguration.json"

        # get the drive letter from the path
        $driveLetter = $logPath.Substring(0,$logPath.IndexOf(":")+1)

        # get the disk free space
        $startFreeSpace = (Get-WmiObject -Query ("SELECT * FROM Win32_LogicalDisk WHERE DeviceID
= '" + $driveLetter + "'")).FreeSpace

        # display  status
        Write-Host " Checking for the presence of a JSON configuration file for the cleanup
process: " -NoNewline

              # if the file exists
              if (Test-Path -Path $logPath)
              {
            # display status
            Write-Host "Found"
```

```
                    # import the JSON file
                    $jsonData = Get-Content -Raw -Path $logPath | ConvertFrom-Json
             }
        else
        {
            Write-Host "Not Found"
        }
       }

    # blank line
    Write-Host ""

       # iterate through them and...
       foreach ($webSite in $webSites)
       {
             # build the name of the folder for this site
             $siteFolder = "W3SVC$($webSite.id)"

             # build the path to the logs for this specific website
             $fullLogPath
=[System.Environment]::ExpandEnvironmentVariables($WebSite.LogFile.Directory) + "\$siteFolder"

             # if the json file is being used then...
             if ($jsonData -ne $null -and $jsonData -ne "")
             {

                    # get the retetnion time for the folder
                    $logRetentionTime =
($jsonData.LogRetentionTimes.$siteFolder.LogRetentionTime)

                    # if the value was null then use a default
                    if ($logRetentionTime -eq $null) { $logRetentionTime = $DefaultLogRetention
}

             }
             else
             {
                    # just use the default
                    $logRetentionTime = $DefaultLogRetention
             }

        # display some status
        Write-Host (" Cleaning Log Files older than " + $logRetentionTime + " days for the Web
Site '" + $webSite.Name + "' (" + $siteFolder + "): ") -NoNewline

             # calculate the date limit
             $dateLimit = (Get-Date).AddDays(-$logRetentionTime)

             # delete files older than the $limit.
             Try
        {
            Get-ChildItem -Path $fullLogPath -Filter "*.log" -Force | Where-Object {
!$_.PSIsContainer -and $_.CreationTime -lt $dateLimit} | Remove-Item -Force
           }

        Catch
        {
            Write-Host "Failed"
            Write-Host " An error was encountered whilst trying to delete the old log files.
The error description is:"
            Write-Host (" " + $_.Exception.Message)
            Write-Host ""
        }

        Finally
        {
            Write-Host "OK"
        }
    }

    # get the disk free space
    $endFreeSpace = (Get-WmiObject -Query ("SELECT * FROM Win32_LogicalDisk WHERE DeviceID = '"
+ $driveLetter + "'")).FreeSpace

    # and write out the saving
    Write-Host (" Total Space Recovered: " + (($endFreeSpace-
$startFreeSpace)/1024/1024).ToString("0.00") + " MB")

    # remove the module
    Remove-Module WebAdministration
}
else
{
       # display status
       Write-Host "Failed"
```

```
        Write-Host ""
        Write-Host "The WebAdministration Module was missing which most likely means that this
Server does not have an implementation of IIS installed."
        Write-Host "The script will now terminate."
}

# write out a blank line and status to finish
Write-Host ""
Write-Host "Cleanup of IIS Log Files Completed"
Write-Host ""
```