Vergleich der beiden Skripte

|  |  |
| --- | --- |
| Werreports\_10.ps1 | Reduz.ps1 |
| function GetUsers  {  return Get-ChildItem "C:\Users" | Select-Object Name  } | function GetUsers  {  $Users = [System.Collections.ArrayList]::New()    $tempUser = Get-ChildItem "C:\Users"|Select-Object Name  foreach($username in $tempUser)  {  #Uername ist ein Objekt mit der Eigenschaft Name  #Write-host "bla " + $username.Name  [User]$u = [User]::New($username.Name)  $Users.Add($u)  }  return $Users  } |

|  |  |
| --- | --- |
| GetReportData() | GetReportData() |
| function GetReportData  {  $stats=@()  $apps=@()    $\_users = GetUsers;    foreach($\_user in $\_users) {  $paths = GetWERPath $\_user.Name;    foreach($\_path in $paths) {  if($\_path -ne $null)  {  $\_reportid = GetReportInnerData $\_path "ReportIdentifier";  $\_reporttype = GetReportInnerData $\_path "ReportType";  $\_eventtype = GetReportInnerData $\_path "EventType"; # z.B AppCrash  $\_eventtime = GetReportInnerData $\_path "EventTime";  $\_bucketid = GetReportInnerData $\_path "Response.BucketId";  $\_appname = GetReportInnerData $\_path "AppName";    $stats += New-Object PSObject -Property @{user=$\_user.Name;report\_id=$\_reportid;report\_type=$\_reporttype;event\_type=$\_eventtype;event\_time=$\_eventtime;bucket\_id=$\_bucketid;app=$\_appname};    $apps += New-Object PSObject -Property @{app=$\_appname};  }  }  }      ### CSV für die anzahl der Fehler pro Benutzer  $csv=@()    $\_users = GetUsers;  foreach($\_user in $\_users) # $\_users hier stehen alle benuter vom Rechner und werden nach $\_user geschrieben in der Schleife  {  $\_tempuser = $\_user.Name; # Temp Benutzername damit er aus der Hashtable genommen werden kann  $\_count = GetWerError $\_tempuser $\_showNames; # Zählt die WER Fehler / Dokumente  $csv += New-Object PSObject -Property @{Benutzer=$\_tempuser;Fehler=$\_count}  $csv | Select-Object Benutzer,Fehler | Export-Csv -Path c:\Benutzer-Fehler.csv -Encoding utf8 -NoTypeInformation -Delimiter ";"  }    ##### Reporttypen pro Nutzer  $\_counter1=0  $\_counter2=0  $\_counter3=0    $csv=@()    foreach($\_user in $\_users) # $\_users hier stehen alle benuter vom Rechner und werden nach $\_user geschrieben in der Schleife  {  $\_tempuser = $\_user.Name; # Temp Benutzername damit er aus der Hashtable genommen werden kann  foreach ($\_error in $stats)  {  if($\_tempuser -eq $\_error.user) #Username an erster Stelle im Array  {  switch ($\_error.report\_type)  {  "1" {$\_counter1 ++}  "2" {$\_conuter2 ++}  "3" {$\_counter3 ++}  }  }  }    $csv += New-Object PSObject -Property @{Benutzer=$\_tempuser;Report\_ID1=$\_counter1;Report\_ID2=$\_counter2;Report\_ID3=$\_counter3}  $csv | Select-Object Benutzer,Report\_ID1,Report\_ID2,Report\_ID3 | Export-Csv -Path c:\Benutzer-Report.csv -Encoding utf8 -NoTypeInformation -Delimiter ";"  }    ### Abfragen der Fehler pro Anwendung  $\_counter1=0  $\_counter2=0  $\_counter3=0    $csv=@()    $apps = $apps | group-object app -noelement | select-object name | sort-object name    foreach($\_app in $apps) # $\_users hier stehen alle benuter vom Rechner und werden nach $\_user geschrieben in der Schleife  {  foreach ($\_error in $stats)  {  if($\_app.Name -eq $\_error.app) #Username an erster Stelle im Array  {  switch ($\_error.report\_type)  {  "1" {$\_counter1 ++}  "2" {$\_conuter2 ++}  "3" {$\_counter3 ++}  }  }  }    $csv += New-Object PSObject -Property @{Anwendung=$\_app.Name;Report\_ID1=$\_counter1;Report\_ID2=$\_counter2;Report\_ID3=$\_counter3}  $csv | Select-Object Anwendung,Report\_ID1,Report\_ID2,Report\_ID3 | Export-Csv -Path c:\Benutzer-AppReport.csv -Encoding utf8 -NoTypeInformation -Delimiter ";"  }  } | function GetReportData  {  # $stats=@()  #$apps=@()    $\_users = GetUsers;    foreach($\_user in $\_users) {  $paths = GetWERPath $\_user.Name;  foreach($\_path in $paths) {  if($\_path -ne $null)  {  $\_reportid = GetReportInnerData $\_path "ReportIdentifier";  $\_reporttype = GetReportInnerData $\_path "ReportType";  $\_eventtype = GetReportInnerData $\_path "EventType"; # z.B AppCrash  $\_eventtime = GetReportInnerData $\_path "EventTime";  $\_bucketid = GetReportInnerData $\_path "Response.BucketId";  $\_appname = GetReportInnerData $\_path "AppName";    [Report]$rep = [Report]::new($\_reportid, $\_reporttype, $\_eventtype, $\_eventtime, $\_bucketid, $\_appname)  $\_user.Reports +=$rep  }  }  }  $\_users  } |
|  | class User  {  <#  .Synopsis  test  .Description  Hält die Informationen pro User  Mehere Reports sind möglich  #>  [string]$Name  #generische Liste  $Reports = [System.Collections.Generic.List[Report]]::New()  [Computer]$pc  #eine ArrayList  #$Liste = [System.Collections.ArrayList]::New();    User([string]$uname)  {  $this.Name = $uname  $this.pc = [Computer]::new()  }  }  class Report  {  [string]$ReportID  [int]$ReportType  [string]$EventType  [string]$EventTime  [string]$BucketID  [string]$Appname  Report($repid, $repType, $evType, $evTime, $buckId, $appnam)  {  $this.ReportID = $repid  $this.ReportType = $repType  $this.EventType = $evTime  $this.EventTime = $evTime  $this.BucketID = $buckId  $this.Appname = $appnam  }  }  class Computer  {  [string]$mac  [string]$OpSys  [string]$Name  Computer()  {  $macadresse = get-wmiobject -class "Win32\_NetworkAdapterConfiguration" | Where {$\_.IpEnabled -Match "True"} | Select MacAddress  $this.mac = $macadresse[0].MacAddress  $this.OpSys = (Get-WmiObject Win32\_OperatingSystem).Name  $this.Name = (Get-WmiObject Win32\_OperatingSystem).CSName    }  } |
|  |  |