# Fiche 14 : Audit de postes vers une base de données

## 1. La fonction audit

Cette fonction récupère les noms du processeur, de la carte graphique et de la carte réseau, le système et le service pack installés et les adresses Mac et IP des postes interrogés.

```
function audit{
    PROCESS{
    Trap{
        Add-Content  "c:\erreur.txt" "problème de connexion avec $nom" -Force
        Set-Variable skip($true) -Scope 1
        continue
    }
    $obj=New-Object PSObject
    $skip=$false
    #récuperation de la configuration de l'ordinateur
    $nom=$_
    $cpu=Get-WmiObject Win32_Processor -ComputerName $nom -ErrorAction stop | select name -First 1
    if (-not $skip){
        $cg=Get-WmiObject Win32_Videocontroller -ComputerName $nom -ErrorAction stop | Select name
        $os=Get-WmiObject Win32_OperatingSystem -ComputerName $nom -ErrorAction stop |
        Select CSName, ServicePackMajorVersion, Caption
        $cr=Get-WmiObject win32_networkadapterconfiguration -Filter IPEnabled=true -ComputerName $nom -ErrorAction stop |
        Select MACAddress,IPAddress,Description -First 1

        # création de l'objet retourné
        $obj | add-member NoteProperty nom ($os.CSName)
        $obj | add-member NoteProperty cpu ($cpu.name)
        $obj | add-member NoteProperty cg ($cg.name)
        $obj | add-member NoteProperty os ($os.Caption)
        $obj | add-member NoteProperty sp ($os.ServicePackMajorVersion)
        $obj | add-member NoteProperty mac ($cr.MACAddress)
        $obj | add-member NoteProperty ip ($cr.IPAddress[0])    #1ère IP de la carte
        $obj | add-member NoteProperty cr ($cr.Description)

        Write-Output $obj
    }
    }
}
```

Voir fiche N°10 sur les interruptions. La variable skip est affectée avec l'instruction *Set-variable* et le paramètre *–Scope 1* précise le contexte de la variable : bloc d'instructions d'un (1)

Si plusieurs valeurs, retourne la

Si plusieurs cartes, ne retourne que la

Retourne le nom local du

Remarque sur l'objet retourné :
La fonction aurait pu retourner les valeurs trouvées avec la simple instruction *Write-host*, mais comme on le verra par la suite, cette solution est beaucoup moins évolutive pour l'utilisation de la fonction :

```
Write-Host $os.CSName,$cpu.name,$cg.name,$os.Caption,$os.ServicePackMajorVersion,$cr.MACAddress,..
```

## 2. Les différentes manières d'appeler la fonction et d'exploiter son résultat

*Exemple avec le nom du poste dans une chaine de caractère :*

```
"localhost" | audit | Format-List
```

```
Console PowerShell
nom : NEMO
cpu :                Intel(R) Pentium(R) 4 CPU 3.00GHz
cg  : RADEON X300 Series
os  : Microsoft Windows XP Professionnel
sp  : 2
mac : 00:11:11:49:6C:35
ip  : 192.168.10.2
cr  : Intel(R) PRO/100 VE Network Connection - Virtua.
```

*Bien sûr, on peut spécifier une liste de postes et avoir une sortie sous forme de tableau :*

```
"XP01","XP02","XP03" | audit | Format-Table
```

*La liste des noms des postes peut être contenue dans un fichier texte et la sortie placée également dans un fichier texte :*

```
Get-Content "c:\listePc.txt" | audit | Export-Csv "c:\listeAudit.txt"
```

*La liste des noms des postes peut être obtenue en interrogeant le domaine Active Directory :*

```
$s=New-Object system.DirectoryServices.DirectorySearcher
$s.filter="(objectcategory=computer)"
$s.findall() | foreach {$_.properties.name} | audit | Export-Csv "c:\listeAudit.txt"
```

*Idem, mais avec les applets de commandes Quest Software (voir fiche N°8) et une identification au domaine :*

```
$login=get-credential
Get-QADComputer -Credential $login -service "w2s.labo.ig" | foreach {$_.name} | audit |
Export-Csv "c:\listeAudit.txt"
```

## 3. Une nouvelle fonction pour l'exportation du résultat de l'audit dans une base de données

Cette fonction ajoute les informations de l'audit dans la table ordinateur de la base de données bdaudit sous SQLServer.
Voir la fiche N°12 pour l'accès à une base de données à partir de PowerShell.

```
function ExportSQL{
    BEGIN{
        Trap{
        Add-Content  "c:\erreur.txt" "problème de connexion à la base de données" -Force
        break
        }
        #connexion à la base de données
        Set-Location SQLSERVER:\SQL\XP\SQLEXPRESS\Databases\bdaudit -ErrorAction stop

        #requete qui vide la table ordinateur
        $req="delete from ordinateur"
        Invoke-Sqlcmd $req -ErrorAction stop
    }
    PROCESS{
        Trap{
        Add-Content  "c:\erreur.txt" "problème d'ajout dans la base pour $nom" -Force
        continue
        }
        $nom=$_.nom
        $cg=$_.cg
        $sp="service pack "+$_.sp
        $os=$_.os
        $ip=$_.ip
        $mac=$_.mac
        #remplacement de caractères posant probleme pour la concatenation de chaine SQL
        $cpu=$_.cpu.Replace("+"," ")
        $cr=$_.cr.Replace("'","''")

        #insertion des informations dans la base de données
        $req="insert into ordinateur values ('$nom','$cpu','$cg','$os','$sp','$mac','$ip','$cr')"
        Invoke-Sqlcmd $req -ErrorAction stop
    }
}
```

Si la connexion échoue, l'exportation s'arrête.

*Le résultat de l'audit  peut maintenant être exporté vers la base de données bdaudit :*

```
Get-Content "c:\listePc.txt" | audit | ExportSql
```

*Ou en interrogeant le domaine Active Directory  :*

```
$s=New-Object system.DirectoryServices.DirectorySearcher
$s.filter="(objectcategory=computer)"
$s.findall() | foreach {$_.properties.name} | audit | ExportSql
```