

Ajouter avec Powershell le service Windows Deployment Share

Pour utiliser ce script powershell, il faudra rajouter dans le paramètre le chemin du dossier à créer. Depuis Windows Powershell, il faudra exécuter le script avec le chemin.

Exemple:

Installer le service Windows Deployment sur votre serveur et utiliser le répertoire C:\WDS.

```
PS-WDSUTIL.ps1 -RemInst C:\WDS
```

Retirer le service Windows Deployment sur votre serveur.

```
PS-WDSUTIL.ps1 -Remove
```

Le Script:

```
<#  
.SYNOPSIS  
    Add or remove Windows Deployment Services for Windows Server.  
.DESCRIPTION  
    Add or remove Windows Deployment Services for Windows Server.  
    PS-WDSUTIL take string for define the folder use for initialize server.  
.PARAMETER Name  
.EXAMPLE  
  
PS-WDSUTIL.ps1 -RemInst C:\WDS  
Initialize Windows Deployment Service on server and use C:\WDS folder  
  
.EXAMPLE  
  
PS-WDSUTIL.ps1 -Remove  
Uninitialize Windows Deployment Service on server  
  
.NOTES  
Script Version 1.0  
Tested on:  Windows Server 2016  
           Windows Server 2016  
.LINK  
https://tdelacour.wordpress.com/  
#>  
  
# Script Parameter  
  
    param(  

```

```
[Parameter(Mandatory = $True)]
[ValidateNotNullOrEmpty()]
[string]$RemInst,
[Parameter(Mandatory = $False)]
[switch]$Remove
)

#-----

#region ProductType
# Get ProductType to define the type of the operating system
Try
{
    # Get information via Cim
    $OSQUERY = Get-CimInstance -Namespace ROOT/cimv2 -ClassName
Win32_OperatingSystem
}
Catch
{
    # if Get-CimInstance not work, get information via Wmi
    $OSQUERY = Get-WmiObject -Namespace ROOT/cimv2 -Class
Win32_OperatingSystem
}

# Detection if the script uses on a server
if ($OSQUERY.ProductType -ne "3"){
Write-Output "This script, it's only for a Windows Server"
Write-Output "This script don't be used on Domain Controller"
Break
}

#endregion

# Check if Windows Deployment Service already exists

$WDSservice = get-service -Name WDSservice

#If the Windows Deployment service is on its way, we will ask you the
question if we withdraw the service

If(($($WDSservice.Status) -eq $null -or $($WDSservice.Status) -eq "Stopped")
-eq $false){
    Write-Output "Would do you remove WDS Service ? (Default is No)"
    $ReadHost = Read-Host " ( Y / N ) "
    Switch ( $ReadHost ) {
        Y {Write-Output "Yes, Remove WDS Service"; $RemoveWDS=$true}
        N {Write-Output "No, Keep WDS Service"; $RemoveWDS=$false}
        Default {Write-Output "Default, Keep WDS Service"; $RemoveWDS=$false}
    }
    if($RemoveWDS -eq $true){
        wdsutil /verbose /Uninitialize-server
    }
}
```

```
    }
    else
    {
        Write-output "Sorry, this script do nothing"
        Break
    }
}

#endregion

# Remove the folder to use by the script, if it already exists, with prompt

If((Test-Path $RemInst) -eq $true)
{
    Write-Error "This folder already exist"
    Remove-Item -Path $RemInst -Force -Recurse -Confirm
    If((Test-Path $RemInst) -eq $true)
    {
        Write-Error "The folder is always present"
        Break
    }
}

#endregion

#region WindowsFeature

# Install Windows Feature wds-deployment
Try {
    $ADDWDS = Install-WindowsFeature wds-deployment -IncludeManagementTools
    $ADDWDS.ExitCode.value__
}
Catch
{
    Write-Error $Error
}

If ($ADDWDS.ExitCode.value__ -eq "1003"){Write-Output "WDS Service is
already install - No Change Needed"}

#endregion

#region CreateFolderReminst

# Create Reminst folder
New-Item -Path $RemInst -Force -ItemType directory

#endregion

#region InstallServic
```

```
# Detect if you computer is on domain or workgroup via cim or wmi
Try
{
    # Get information via Cim
    # PartOfDomain (boolean Property)
    $PartOfDomain = Get-CimInstance -Namespace ROOT/cimv2 -ClassName
Win32_ComputerSystem | Select PartOfDomain | ft -AutoSize -HideTableHeaders
    # WorkGroup (String Property)
    $WorkGroup = Get-CimInstance -Namespace ROOT/cimv2 -ClassName
Win32_ComputerSystem | select Workgroup | ft -AutoSize -HideTableHeaders
}
Catch
{
    # if Get-CimInstance not work, get information via Wmi
    # PartOfDomain (boolean Property)
    $PartOfDomain = (Get-WmiObject -Class Win32_ComputerSystem -Namespace
ROOT/cimv2) | select PartOfDomain | ft -AutoSize -HideTableHeaders
    # WorkGroup (String Property)
    $WorkGroup = (Get-WmiObject -Class Win32_ComputerSystem -Namespace
ROOT/cimv2) | select Workgroup | ft -AutoSize -HideTableHeaders
}

# Run WDSUTIL with the right settings, which depends on whether the server
has on domain or not

if (!$PartOfDomain)
{
    Write-Output "Domain"
    WDSUTIL /Verbose /Progress /initialize-server /remInst:$RemInst
}
Else
{
    if (!$WorkGroup)
    {
        Write-Output "Error Wmi\Cim filter"
        break
    }
    Else
    {
        Write-Output "Workgroup"
        WDSUTIL /Verbose /Progress /Initialize-Server /REMINST:$RemInst
/Standalone
    }
}

#endregion
```

From:

<http://poste2travail.free.fr/dokuwiki/> - **Poste2Travail**

Permanent link:

<http://poste2travail.free.fr/dokuwiki/doku.php?id=script:powershell:wds>



Last update: **2020/08/10 23:07**