

Fiche 10 : Interception des erreurs

1. Mise en place de l'interruption

Le script suivant permet de récupérer des informations sur des ordinateurs avec l'objet *get-wmiobject*. Si l'ordinateur ne répond pas, son nom est écrit dans un fichier nommé *c:\temp\erreur.txt*.

```
1 trap {  
2     Add-Content c:\temp\erreur.txt "$computer" -Force  
3     continue  
4 }  
5 $computer = "server2"  
6 get-wmiobject win32_operatingsystem -ComputerName $computer -ErrorAction Stop  
7 $computer = "localhost"  
8 get-wmiobject win32_operatingsystem -ComputerName $computer -ErrorAction Stop  
9 Write-Host "Fin"
```

Force la création du chemin s'il n'existe

Explications :

Lignes 1 à 4 : C'est le code de l'interruption, il est placé en début de script mais ne sera exécuté que si une exception se produit dans le script.

Ligne 3 : Le mot *continue* spécifie de reprendre l'exécution du script à la ligne qui suit celle qui a déclenché l'exception.

Ligne 7 : Le paramètre *-ErrorAction* spécifie le comportement de la commande lorsqu'une erreur se produit, ici on utilise la valeur *Stop* pour intercepter l'erreur et exécuter l'interruption *trap* des lignes 1 à 4.

Déroulement : Si l'ordinateur "server2" n'existe pas ou ne répond pas, la ligne 6 va déclencher une exception, la ligne 2 de l'interruption va écrire le nom du serveur dans le fichier *erreur.txt* et le mot *continue* de la ligne 3 va permettre de reprendre le déroulement du script à la ligne 7.

Remarque :

- La détection de la non réponse de l'ordinateur peut-être assez longue, donc l'écriture dans le fichier *erreur.txt* n'est pas immédiate.

2. Interception de toutes les erreurs

Lorsqu'une erreur se produit, l'interpréteur de commande examine la variable PowerShell *\$ErrorActionPreference* qui par défaut a la valeur "*continue*". En affectant la valeur "*Stop*" à cette variable, l'interruption *trap* sera aussi exécutée, mais pour toutes les erreurs dans le déroulement du script, ce qui n'est pas des plus pratiques.

```
1 $ErrorActionPreference = "Stop"  
2 trap {  
3     Add-Content c:\temp\erreur.txt "$computer" -Force  
4     continue  
5 }  
6 $computer = "server2"  
7 get-wmiobject win32_operatingsystem -ComputerName $computer
```

Ligne 7 : Le paramètre *-ErrorAction* n'est plus nécessaire.

3. Les différentes actions possibles en cas d'erreur

Les valeurs possibles de la variable *\$ErrorActionPreference* ou du paramètre *-ErrorAction* sont :

Continue : (valeur par défaut), un message d'erreur s'affiche et le script continue.
SilentlyContinue : Le script continue sans afficher de message d'erreur.
Inquire : Un message d'erreur s'affiche avec une demande à l'utilisateur pour continuer.
Stop : Un message d'erreur s'affiche et le script s'arrête ou si elle existe, l'instruction *trap* est lancée.