

## Automatisation des tâches Builder et Integrator – Astuces (2ème partie)

### Description

Dans [l'astuce du 5 décembre 2011](#), nous avons indiqué comment consigner dans un fichier texte les logs d'exécution de Builder et Integrator lorsqu'on utilise des fichiers batch (.bat) et des tâches planifiées de Windows®.

Nous allons voir cette fois-ci comment consigner ces même logs en utilisant des scripts VBS (.vbs) et des tâches planifiées de Windows®.

Supposons que l'on veuille exécuter la ligne de commande suivante :

```
C:\DI-Solution\Executables\integ.exe C:\DI-Projects\Programs\script1.int
```

Pour pouvoir l'exécuter à partir d'un script VBS, dans le fichier .vbs la syntaxe doit être la suivante :

```
Set WshShell = WScript.CreateObject("WScript.Shell")  
stCde="C:\DI-Solution\Executables\integ.exe C:\DI-Projects\Programs\script1.int"  
WshShell.Run stCde
```

Si maintenant, vous souhaitez récupérer les logs générés lors de l'exécution du script il faut utiliser les lignes de code suivantes :

```
Set WshShell = WScript.CreateObject("WScript.Shell")  
stCde="%COMSPEC% /c C:\DI-Solution\Executables\integ.exe C:\DI-Projects\Programs\script1.int  
>> C:\DI-Projects\Logs\script1.jou"  
WshShell.Run stCde
```

(ajoute les logs au fichier **script1.jou** existant)

ou bien celles-ci :

```
Set WshShell = WScript.CreateObject("WScript.Shell")  
stCde="%COMSPEC% /c C:\DI-Solution\Executables\integ.exe C:\DI-Projects\Programs\script1.int  
> C:\DI-Projects\Logs\script1.jou"  
WshShell.Run stCde
```

(supprime d'abord le contenu du fichier **script1.jou** puis ajoute ensuite les logs en cours)

La deuxième syntaxe évite d'avoir un fichier **script1.jou** dont la taille augmente au fil du temps.

### Tags

1. automatisation
2. Builder
3. Data Integrator
4. script