



1

Introduction à Windows Script Host

1. Introduction à Windows Script Host

Windows Script Host (*WSH*) est un environnement permettant d'héberger et d'exécuter des scripts afin d'automatiser des tâches administratives sur l'ordinateur. Avant Windows 98, aucun outil n'existait pour accomplir de telles tâches si ce n'est que le Macro Recorder de Windows 3.1 et les fichiers Batch (*.bat) typiques à DOS. Cependant, ces derniers possédaient de fortes limitations tant qu'au niveau de la structure du langage qu'au niveau des fonctionnalités accessibles.

Description et applications de Windows Script Host

Lorsqu'un script s'exécute sur votre ordinateur, WSH joue le rôle de l'hôte en lui rendant accessibles une pluralité d'objets et de services ainsi qu'une série de règles au sein desquelles le script est exécuté. De plus, *Windows Script Host* gère l'aspect sécurité entourant l'exécution d'un script et invoque le moteur de script approprié.

Windows Script Host est l'idéal pour l'administrateur qui désire créer des scripts comme des scripts de connexion et d'administration automatisée. WSH vous permet d'accomplir des tâches comme :

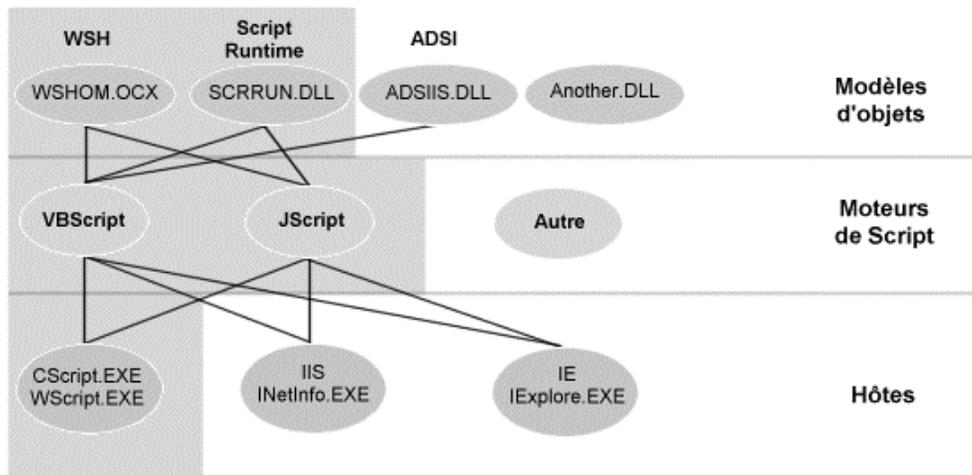
- Imprimer des messages à l'écran
- Exécuter des fonctions de base
- Créer des icônes, démarrer des logiciels
- Effectuer des sauvegardes (*backup*)
- Connecter des lecteurs réseau
- Connecter des imprimantes
- Récupérer ou modifier des variables d'environnement
- Modifier des clés de la base de registres
- Ajouter, modifier et supprimer des utilisateurs NT
- Redémarrer, verrouiller des stations de travail

Windows Script Host est intrinsèque à Microsoft Windows 98, 2000, and Millennium. Cependant, l'installation de Windows 98 ne provoque pas l'installation de *Windows Script Host*. Vous devez donc l'installer manuellement. Si vous utilisez Windows 95 ou NT 4.0, vous devez posséder *Internet Explorer 4* ou plus et vous pouvez télécharger *Windows Script Host* à partir du site web de Microsoft à l'adresse <http://msdn.microsoft.com/scripting>. Sous Windows NT 4.0, l'installation de l'*Option Pack 4* remédie à la situation.

Architecture de Windows Script Host

Lors de l'exécution d'un script, trois différentes couches de composants sont sollicitées.

1. L'exécution d'un script est d'abord lancée par un **hôte** qui charge le moteur de script approprié au langage utilisé au sein du script.
2. Le **moteur de script** précédemment chargé interprète le script et invoque les différentes fonctions utilisées. Ces fonctions sont stockées au sein de fichiers binaires et sont regroupées au sein de modèles d'objets.
3. Les **modèles d'objets** contiennent les fonctionnalités nécessaires à l'exécution de tâches administratives.



Hôtes

Les hôtes sont les programmes qui exécutent les scripts au sein de leur propre processus. Bien avant WSH, les scripts pouvaient être exécutés par *IIS* et *Internet Explorer* ce qui réduisait l'utilisation des scripts à l'environnement web. Désormais, grâce à *CScript.exe* et *WScript.exe*, les scripts peuvent être exécutés à l'extérieur du web.

Moteurs de scripts

La grande majorité des scripts sont écrits à l'aide de Microsoft JScript ou VBScript puisque ces moteurs de script sont implantés au sein même de Microsoft Windows 98, 2000 et Millennium. Cependant, il est possible d'utiliser d'autres moteurs de scripts tels Perl, REXX et Python au sein du même environnement *Windows Script Host*.

Lorsque l'hôte a déterminé quel langage est utilisé au sein du script, il charge le moteur de script approprié et lui passe le script afin d'être exécuté. Notez que le moteur de script n'a pour tâche que d'interpréter le script et d'invoquer l'exécution des fonctions appropriées stockées au sein de fichiers binaires que sont les modèles d'objets.

Un fichier de script écrit à l'aide du langage JScript devra posséder l'extension **.js* tandis qu'un fichier de script écrit à l'aide du langage VBScript devra posséder l'extension **.vbs*. Ces extensions sont enregistrés au sein de la base de registres de *Windows* ce qui en permet l'exécution simplement en double-cliquant sur le fichier. Lorsque vous activez ces types de



fichiers, *Windows* démarre *Windows Script Host* qui, ce dernier, se charge d'invoquer le moteur de script approprié afin d'interpréter le script.

Modèles d'objets

Les scripts utilisent des objets prédéfinis. Un objet peut être perçu comme un lot de fonctionnalités regroupés thématiquement sous une appellation nommée *Modèle d'objets*. Vous utilisez les objets au sein de vos scripts afin d'accéder aux fonctionnalités de l'ordinateur, de la base de registre, du réseau, etc. Ces objets sont compilés au sein de fichiers binaires aux extensions *.ocx ou *.dll et seront généralement stockés dans le répertoire système de l'ordinateur.



WSH est grandement extensible. Vous pouvez donc installer des moteurs de script et des modèles d'objets supplémentaires. Cependant, les autres moteurs de script que VBScript et JScript et les autres modèles d'objets doivent préalablement être adéquatement enregistrés au sein de la base de registre avant d'être utilisés.

Création et exécution des scripts

Un script Windows est un fichier texte standard. Vous pouvez créer un script à l'aide de n'importe quel éditeur de texte tant que celui-ci n'insère pas de balises de mise en forme du texte tel que *Winword* ou *Wordpad*. Ainsi, le plus répandu des éditeurs de texte standard demeure Bloc-notes (*Notepad*). Afin d'être adéquatement reconnu par WSH, le fichier devra posséder une des extensions compatibles (**.js*, **.vbs*, or **.wsf*).

Création d'un script à l'aide de Bloc-notes

1. Démarrez Bloc-notes (*Notepad*).
2. Écrivez le texte suivant :

```
WScript.Echo "Allô la planète!"
```

CH01\Allo.vbs

3. Sauvegardez ce fichier en lui donnant l'extension **.vbs* à la place de l'extension **.txt* insérée par défaut par Bloc-notes. Vous créez ainsi un fichier qui devra être interprété par le moteur de scripts VBScript.
4. Localisez le fichier que vous venez de créer et activez-le en double-cliquant sur son icône. *Windows Script Host* invoque le moteur VBScript qui exécute votre script. La boîte de message suivante s'affiche alors à l'utilisateur:



Exécution d'un script à l'invite de commande

WScript.exe est l'hôte utilisé par défaut par Windows pour exécuter les scripts. C'est donc cet hôte qui est invoqué automatiquement lorsque vous activez un script en double-cliquant sur son icône. Vous pouvez cependant utiliser un autre hôte, *CScript.exe*, qui est une version à l'invite de commande de *WScript.exe* et qui peut être utilisée pour lancer l'exécution de scripts à partir de fichiers Batch (**.bat*).

```
cscript [options de l'hôte...] [nom du script] [paramètres du script]
```

- **Options de l'hôte** permet d'activer ou de désactiver un certain nombre de fonctionnalités Windows Script Host. Les options de l'hôte sont précédées de deux barres obliques (*//*).
- **Nom du script** est le nom du fichier de script avec son extension et les informations de chemin appropriées, par exemple, *d:\admin\vbscripts\chart.vbs*.
- **Les paramètres du script** sont transmis au script en tant que ligne de commande. Les paramètres destinés au script sont précédés d'une seule barre oblique (*/*).

Exemples de l'exécution de scripts à l'invite de commande :

```
cscript //logo c:\scripts\script.vbs
cscript //nologo //B "c:\sample scripts\script.vbs"
```

Voici la liste complète des différents options de l'hôte que vous pouvez utiliser lors de l'exécution de scripts à l'invite de commande.

Option	Description
//B	Mode Batch. Supprime l'affichage d'invites utilisateur et d'erreurs de script sur la ligne de commande. Le mode par défaut est le mode Interactive.
//D	Active le débogueur. Voyez plus loin dans ce chapitre les explications au sujet de l'utilisation du débogueur.
//E: <i>moteur</i>	Exécute le script avec le moteur de script spécifié.
//H:Cscript	Définit <i>CScript.exe</i> comme l'application par défaut pour l'exécution de scripts pour la durée de la session active de l'utilisateur. Spécifiez l'option //S si vous désirez conserver cette valeur pour les prochaines sessions de l'utilisateur.
//H:Wscript	Définit <i>WScript.exe</i> comme l'application par défaut pour l'exécution de scripts pour la durée de la session active de l'utilisateur. Spécifiez l'option //S si vous désirez conserver cette valeur pour les prochaines sessions de l'utilisateur.
//I	Mode Interactif ; permet d'afficher les invites utilisateur et les erreurs de script. Opposé du mode Batch.
//Job: <i>JobID</i>	Exécute le numéro d'ID de tâche spécifié à partir du fichier *.wsf.
//logo	Affiche une bannière. Opposé de <code>nologo</code> .
//nologo	Empêche l'affichage d'une bannière d'exécution lors de l'exécution. Le paramètre par défaut est <code>logo</code> .
//S	Enregistre les options de ligne de commande actuelles et les définit par défaut pour cet utilisateur.
//T: <i>nn</i>	Active le délai d'expiration : le nombre maximal de secondes pendant lesquelles le script peut être exécuté au delà de lesquelles le script doit être interrompu. Ce paramètre empêche des délais d'exécution excessifs à l'aide d'une horloge. La valeur par défaut est illimitée.
//U	Utilisé avec Windows NT et Windows 2000 pour forcer la sortie de ligne de commande à utiliser les caractères Unicode. <i>CScript</i> est incapable de déterminer si la sortie doit être en Unicode ou en ANSI ; par défaut, il produit la sortie en ANSI.
//X	Lance le programme dans le débogueur. Voyez plus loin dans ce chapitre les explications au sujet de l'utilisation du débogueur.
//?	Affiche une courte description des paramètres de commande ainsi que des informations d'utilisation.

Utilisation du débogueur

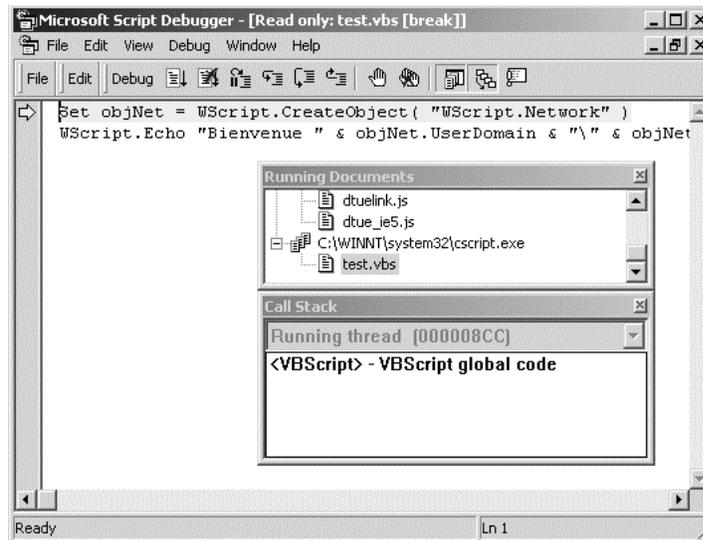
Lorsque vous lancez l'exécution de votre script à partir de l'invite de commande en combinaison avec les options `//D` et `//X`, le débogueur est activé et le script lancé y est chargé afin d'être exécuté pas-à-pas. Le débogueur vous permettra de suivre étapes après étapes l'exécution de vos scripts afin d'en localiser plus aisément les erreurs.

Microsoft Script Debugger doit être installé sur votre ordinateur afin de pouvoir déboguer vos scripts. *Microsoft Script Debugger* est intégré aux différentes versions de Windows 2000 mais ne fait pas partie des composants s'installant par défaut lors de l'installation du système d'exploitation. Pour l'installer, vous devez activer **Démarrer** → **Paramètres** → **Panneau de configuration** → **Ajout/Suppression de programmes** → **Ajouter/Supprimer des composants Windows**. Sous les autres versions de Windows, vous devez installer manuellement *Microsoft Script Debugger* qu'il est possible de télécharger gratuitement à partir de l'adresse suivante :

<http://msdn.microsoft.com/downloads/default.asp?url=/downloads/topic.asp?url=/msdn-files/028/001/175/topic.xml>

Dès le lancement de l'exécution de votre script, *Microsoft Script Debugger* s'affiche et met en surbrillance la ligne qu'il s'apprête à exécuter.

Plusieurs fenêtres peuvent être affichées afin de visualiser certaines informations utiles dont le numéro assigné par Windows au thread courant (*était-ce vraiment nécessaire Bill ?*)



Maintenant, si vous appuyez sur la touche F8, vous verrez l'exécution de votre code s'exécuter pas-à-pas. Notez que la ligne de code en surbrillance est la ligne qui s'apprête à être exécutée et non celle qui vient de l'être.



Lorsqu'une erreur survient au sein d'un script, une boîte de message demande à l'utilisateur s'il désire déboguer le script. *Microsoft InterDev* est alors automatiquement exécuté lorsque celui-ci est installé sur l'ordinateur. Ceux qui veulent modifier ce comportement par défaut et exiger l'utilisation de *Microsoft Script Debugger* afin de déboguer les scripts erronés doivent modifier la base de registres et assigner la valeur `C:\Program Files\Microsoft Script Debugger\mscrdbg.exe` à la clé `HKEY_CLASSES_ROOT\CLSID\{834128A2-51F4-11D0-8F20-00805F2CD064}\LocalServer32`.

Si vous désirez revenir à l'utilisation de Visual InterDev, modifiez la clé pour lui assigner la valeur `C:\Program Files\Microsoft Script Debugger\mdm.exe`.

Icône	Raccourci	Description
	(F5)	Interrompt l'exécution pas-à-pas afin de continuer l'exécution normale du script.
	(Shift + F5)	Interrompt le débogueur ainsi que l'exécution du script.
		Interrompt l'exécution pas-à-pas afin de continuer l'exécution normale du script jusqu'au prochain point d'arrêt.
	(F8)	Continue l'exécution pas-à-pas en entrant dans le code des sous-procédures appelées si celles-ci n'appartiennent pas à un code compilé.
	(Shift + F8)	Continue l'exécution pas-à-pas en n'entrant pas dans le code des sous-procédures appelées même si celles-ci n'appartiennent pas à un code compilé.
	(Ctrl + Shift + F8)	Continue l'exécution pas-à-pas en quittant le code d'une sous-procédure.
	(F9)	Place un point d'arrêt sur la ligne pointée. Ainsi, lors d'une exécution continue du script, un arrêt de l'exécution sera provoquée avant que la ligne pointée ne soit exécutée. Permet également de supprimer un point d'arrêt existant sur la ligne pointée.
	(Ctrl + Shift + F9)	Supprime l'ensemble des points d'arrêt précédemment créés.