

1

Scripting Windows, pour quoi faire ?

Dans ce premier chapitre, nous allons répondre aux premières questions qui doivent naturellement émerger : « Quel est l'intérêt du scripting, et dans quel cadre l'utilise-t-on ? ». Cette première mise en jambe va vous permettre de voir l'utilité au quotidien de scripting dans le cadre de la gestion d'infrastructures système.

Dans quel contexte ?

Qu'est-ce qu'un script ? Un script est un fichier au format texte comprenant un ensemble de commandes écrites dans un langage interprété s'exécutant sur un système d'exploitation. Penchons-nous tout d'abord sur les raisons d'être du scripting, et découvrons son utilité dans la gestion quotidienne d'infrastructure système. Nous allons trouver ici les bonnes raisons qui vous pousseront à abandonner un instant la souris et les icônes au profit d'un éditeur de texte.

La mauvaise réputation

À notre époque de convivialité visuelle, où administration et gestion d'infrastructure système sont synonymes d'interfaces graphiques et de manipulations à la souris, parler de scripting en environnement Microsoft éveille chez les non-initiés un sentiment désagréable de retour arrière, voire d'anachronisme.

« Pourquoi se compliquer la vie avec du code alors qu'on a mis des années à obtenir des interfaces conviviales ? »

« Je mets bien moins de temps à utiliser les outils disponibles plutôt que réfléchir à un script. »

« Je n'ai jamais eu besoin de scripts jusqu'à aujourd'hui, pourquoi cela changerait-il ? »

Voici un cocktail représentatif des remarques que nous entendons le plus souvent dès que nous parlons scripts avec nos collègues ou clients.

C'est un fait : le scripting a encore aujourd'hui une mauvaise image ! Script rime avec programmation, complexité, manque d'intuitivité pour beaucoup d'administrateurs système, exploitants et consultants.

Et pourtant ! La plupart de ces a priori sont en fait loin de la vérité. S'il est vrai qu'il faut apprendre un langage de programmation pour scripter, celui-ci est très simple dans sa structure, suffisamment pour que des profils orientés système et n'ayant pas d'affinité avec le développement (voire une pointe d'hostilité), puissent s'y plonger rapidement et ne plus s'en passer une fois le cap de l'initiation franchi.

Mais alors, quels sont les arguments irréfutables pour pousser n'importe quel profil technique en relation avec l'administration système vers les joies du *VBScript* ?

Les limites des interfaces graphiques

Pour la petite gestion, comme la création d'un compte utilisateur, ou le changement d'un mot de passe (pour prendre les exemples les plus courants), les outils d'administration par défaut font l'affaire : j'ouvre l'outil, je clique sur l'élément, je change un attribut et hop ! Tout est fait.

Malheureusement pour nous, ce genre de manipulations ne se fait pas toujours à petite échelle. Que se passe-t-il quand on veut faire une modification massive sur les utilisateurs ? Renommer un grand nombre de partages ? Faire des modifications dans la base de registre sur la totalité des stations de travail de l'entreprise ?

Là bien sûr, les choses se corsent : à moins de désigner une équipe conséquente à temps plein pendant quelques jours ou acheter un outil d'administration spécifique, il n'y a généralement pas de solution toute prête.

L'argument est irréfutable : les interfaces graphiques ne couvrent pas l'ensemble des cas de figure qui se présentent aux administrateurs de manière satisfaisante, et ce, quelle que soit la taille du réseau à maintenir.

C'est là que le scripting entre en scène.

Les cas courants où le scripting facilite la vie

On distingue deux grandes catégories de travaux où le scripting a un avantage certain sur les outils classiques :

- Automatiser des tâches répétitives.
- Accéder à des fonctionnalités du système qui ne sont pas directement accessibles via les interfaces graphiques (très nombreuses au demeurant).

On rencontre aussi des cas où ces deux raisons se mêlent : automatiser des fonctionnalités *cachées* n'est pas un cas exceptionnel.

Par exemple, on peut très bien imaginer un script permettant de connaître le numéro de version du BIOS d'un parc de machines : ce script est d'ailleurs très facile à faire. Imaginez maintenant la même chose à effectuer manuellement pour 500, 1 000, voire 10 000 machines !

Nous verrons dans ce livre un nombre important de problématiques où le scripting permet de trouver une solution avec une certaine décontraction par rapport aux fanatiques de l'interface.

Améliorer la productivité côté administration et support technique

L'administration et la maintenance d'infrastructure système amènent à effectuer bon nombre de manipulations qui peuvent être directement exploitables dans des scripts.

On retiendra particulièrement :

- la gestion des comptes utilisateurs, machines, groupes de sécurité ;
- la gestion réseau (DNS, DHCP, WINS, etc.) ;
- la maintenance des services (installation/désinstallation, configuration, comptes utilisés, etc.) ;
- l'audit de l'existant ;
- la gestion globale d'*Active Directory* pour les systèmes Windows versions 2000 et 2003 ;
- reporting pour les serveurs et postes de travail (audit des imprimantes, espaces disques, charge des processeurs, boîtes e-mail, etc.).

Le panel des actions scriptables au niveau administration système est très vaste : nous pouvons avancer sans trop de risque que l'ensemble des composants du système sont accessibles et gérables par script.

Le gain d'efficacité lié à l'apprentissage du scripting prend ici tout son sens : que celui qui n'a jamais été confronté à la réalisation d'actions répétitives régulières pour l'administration des utilisateurs, des stations de travail et des serveurs d'entreprise nous jette la première souris sans fil.

Deuxième point fort du scripting : il vous permet de bien comprendre le fonctionnement de votre système. En apprenant le scripting, vous irez plus profondément dans le système en découvrant ses véritables mécanismes. Vous vous apercevrez que les interfaces graphiques obéissent à une logique qui n'est pas nécessairement celle du système proprement dit. Comme pour une voiture, il est plus facile d'appréhender une panne quand on comprend le mécanisme interne du moteur : les vraies causes sont plus facilement décelables.

Sur le terrain, la différence de perception d'une problématique entre un administrateur courant et un initié au scripting est assez flagrante : les problèmes sont pris sous un autre angle et le panel de solutions est beaucoup plus étoffé.

Lors de nos missions orientées scripting, nous ne comptons plus le nombre de fois où nous avons entendu dire « Déjà ? Et dire qu'on cherchait une solution depuis des semaines ». Ce qui pousse généralement le personnel technique à s'intéresser fortement à la chose. Les administrateurs de système Unix/Linux connaissent parfaitement l'importance du scripting qui fait partie intégrante de l'administration quotidienne des serveurs, la tendance aujourd'hui est la même pour les systèmes Windows, chaque nouvelle version développant considérablement cette approche du système.

Ce livre étant principalement axé sur ce sujet, vous découvrirez au fil des chapitres l'ensemble des problématiques systèmes auquel le scripting apporte une réponse.

Améliorer la productivité côté poste de travail

Le scripting n'est pas exclusivement réservé à la gestion *back-office* (applications d'administration des données fournies par les transactions de type commercial réalisées par le *front-office*). On peut même dire que tout l'environnement utilisateur peut profiter de vos créations. En particulier :

- la gestion des scripts de connexions ;
- l'automatisation d'applications bureautique ;
- la maintenance du système d'exploitation ;
- la personnalisation de l'interface ;
- l'aide pour effectuer des tâches complexes pour les utilisateurs courants.