

Réseau Kwartz

Adapter le 'logon script' du serveur

Qu'est-ce que le 'logon script' ?

Le 'logon script', ou script de connexion, est un ensemble de commandes intégrées dans le fichier 'logon.bat' du dossier caché 'netlogon' du serveur. Ces commandes seront lues et exécutées lors de l'ouverture de session d'un usager sur le domaine géré par le serveur Kwartz.

Il est important de préciser que si l'utilisateur n'ouvre pas de session sur le domaine, ce fichier ne sera pas exécuté.

Il exploite les commandes disponibles au sein de l'interpréteur de commande de Windows.

Détail sur les variables d'environnement pouvant être utilisées dans le 'logon script'

Une variable d'environnement est un paramètre initialisé lors de l'ouverture de session sur le domaine et qui contient une information spécifique (concernant les aspects matériel, utilisateur, système, etc.).

Ces paramètres peuvent être alors utilisés au sein du fichier de commande et exécutés à l'ouverture de session.

Parmi ces variables on trouve :

Nom de la variable	Contenu
arch	Mémorise la version du système d'exploitation installé sur le poste se connectant. Les valeurs possibles sont : <ul style="list-style-type: none"> - Vista : Windows Vista, 7, 8 - WinXP : Windows XP - WinXP64 : Windows XP x64 - Win2K3 : Windows 2003 - Win2K : Windows 2000 - WinNT : Windows NT - Win95 : Windows 9x/Millénium - WfWg : Windows 3.11 for workgroup - Samba : Client Linux
server	Mémorise le nom réseau (Netbios) du serveur Kwartz.
client	Mémorise le nom réseau (Netbios) du poste se connectant.
group	Mémorise le groupe d'affectation de l'utilisateur se connectant au domaine.
user	Mémorise le nom du compte utilisateur se connectant au domaine.
clientgroup	Mémorise le nom du groupe de postes de la machine se connectant.
clientip	Mémorise l'adresse IP du poste se connectant.

Comment utiliser les variables d'environnement ?

Il est possible d'utiliser tous les programmes et outils en ligne de commande qui exploitent ces variables.

Pour faire référence au contenu d'une variable on utilisera la syntaxe '%Variable%'.

Par exemple %arch% sera la valeur stockée dans la variable 'arch' concernant le système d'exploitation du poste se connectant.

La commande en ligne 'echo %arch' affichera par exemple 'Vista' pour un poste sous Windows 7, Vista ou 8.

Sous Windows, on pourra obtenir une aide en ligne sur une commande particulière en utilisant la syntaxe 'Help commande' (par exemple 'help if'). La commande 'help' seule affiche les diverses commandes disponibles de l'interpréteur.

Mise en oeuvre

Le fichier 'logon.bat' est un ensemble de lignes de commandes qui s'exécute de manière séquentielle (ligne après ligne). Ces commandes peuvent être listées en exécutant la commande 'help' au sein d'une console.

Nota : on ouvre une console de commande en exécutant au sein de Windows l'application 'cmd' en mode administrateur.

Vous trouverez ci-dessous quelques commandes parmi les plus usitées dans le fichier 'logon.bat'.

a) Le saut conditionnel

Cette commande permet d'effectuer un saut conditionnel dans l'ensemble des lignes de commande.

On utilise pour cela la commande : *IF Condition Goto Etiquette*

- *Condition* représente une expression booléenne dont le résultat est vrai ou faux

- *Etiquette* est une étiquette définissant un point de saut.

Exemple de commandes dans le fichier 'logon.bat' :

```
If not "%arch%"=="Win95" goto NotWin9X
Rem commandes exécutées uniquement sous win9X
...
Goto User
:NotWin9X
Rem commandes exécutées sous les autres systèmes
....
:User
```

La variable 'Arch' est initialisée par le serveur préalablement à l'exécution du fichier 'logon.bat'.

On décodera cet ensemble de commandes de la manière suivante :

- Si la variable 'Arch' ne contient pas (la négation est donnée par le 'not' et la comparaison par '==') 'Win95' alors aller à l'étiquette 'NotWin9x'
 - La suite des lignes concerne les systèmes pour Win9x et peut être exécutées.
 - Aller à l'étiquette 'User'
- ```
:NotWin9X
- Exécuter les commandes pour les systèmes différents de Windows 9x
:User
- Suite du traitement effectué quel que soit le système
```

Autre exemple de commandes dans le fichier 'logon.bat' :

```
If not exist drapeau.tx goto FichierAbsent
Rem le fichier 'drapeau.txt' est présent dans le dossier courant
...
Goto SuiteTraitement
:FichierAbsent
Rem le fichier est absent, on crée un fichier vide
Copy nul drapeau.txt
...
:SuiteTraitement
...
```

**Aide** :help if

**b) La boucle de traitement**

Cette commande permet d'effectuer le même traitement sur un ensemble de fichiers identifiés.

Exemple de commandes dans le fichier 'logon.bat' :

```
For %a in (*.txt) do copy .\%a .\%a.sav
```

On récupère les noms des fichiers dont l'extension 'txt' puis on les recopie en ajoutant l'extension 'sav'.

**Aide** :help for

**c) Appel d'un fichier de commande à partir du fichier en cours.**

Permet d'appeler un fichier de commande à partir d'un autre fichier de commande.

Exemple du fichier 'logon.bat' :

```
...
if exist %0\..\%user%.bat call %0\..\%user%.bat
...
```

S'il existe (if exist) dans le dossier 'Netlogon' du serveur (%0\..\) un fichier 'bat' au nom de l'utilisateur (%user%.bat) alors on suspend l'exécution du fichier en cours (logon.bat) puis on exécute le contenu du fichier '%user%.bat'.

A la fin de l'exécution de ce fichier, on reprend les commandes du fichier initial (logon.bat).

Le 'call' permet, contrairement à un appel direct du fichier, de stopper temporairement l'exécution puis d'exécuter les commandes du fichier appelé. A la fin de l'exécution, on reprend à la ligne suivant le 'Call'.

**Aide** : help call

**d) Interruption de l'exécution d'un fichier de commande et attente d'une action.**

Exemple

```
....
Pause Message_a_afficher
```

...

La commande affiche le message 'Message\_a\_afficher' puis attend l'appui sur une touche du clavier. Cette commande pourra vous être utile afin d'afficher un message pour l'utilisateur mais aussi dans le cadre de la mise au point d'un fichier de commandes.

**Aide** : help pause

**e) Commenter son fichier de commandes.**

Exemple

```
....
Rem Affiche le contenu du répertoire 'parent' du répertoire en cours
Dir ..*
```

Rem n'est pas une commande mais permet l'insertion d'un commentaire dans le fichier de commandes. Il est utile de commenter un fichier de commande afin d'indiquer ce que l'on désire faire.

**Aide** : help rem

**f) Définir une variable et son contenu.**

Exemple

```
....
Rem Déclare la variable Dossier et lui affecte la valeur 'tempo'
Set Dossier=tempo
Rem Affiche le message 'Valeur de X ?' puis attend une saisie de l'utilisateur qui sera stocké
Rem dans la variable SaisieUsr
Set /P SaisieUsr="Valeur de X ?"
```

...

'Set' utilisé sans paramètre permet de visualiser l'ensemble des paramètres d'environnement du système. Utilisé avec un paramètre, il permet d'initialiser ce dernier.

**Aide** : help set

Il existe de nombreuses autres commandes au sein de l'interpréteur : commandes internes (copy, ren, etc.). Il faut aussi ajouter à cela un certains nombres de commandes non intégrées dans l'interpréteur mais disponibles dans le système : commandes externes (diskcopy, format, etc.).