

LES FICHIERS BATCH

1 – Définition

BATCH veut dire « lot » en anglais. Un fichier BATCH est un **programme** écrit en mode texte et qui a l'extension « .bat ». Il permet de lancer une ou plusieurs commandes DOS à la suite l'une de l'autre, et ainsi d'**automatiser** des tâches fastidieuses.

Les fichiers BATCH s'écrivent dans un éditeur de texte (NOTEPAD ou EDIT) et s'exécutent ligne par ligne par l'interpréteur de commandes DOS. Appel du fichier BATCH : nom_du_fichier [paramètre(s)]

2 – Le passage de paramètre

Dans la plupart des cas, une commande a toujours besoin d'un paramètre pour s'exécuter.

Exemple :

c:\>md test : la commande MD a besoin du nom de fichier à créer.

Les commandes incluses dans le fichier peuvent être paramétrées, c'est à dire que ces commandes nécessitent des options (**paramètres formels**) dont la valeur peut changer d'une exécution à une autre du fichier batch.

Un fichier de commande est donc composé de :

- commandes
- paramètres

Les paramètres des commandes contenues dans un fichier batch peuvent être :

- Soit fixes
- Soit variables

L'intérêt d'avoir des paramètres variables est de pouvoir réutiliser un même script avec des paramètres différents.

Pour paramétrer les commandes d'un batch avec des valeurs variables, l'utilisateur se sert de paramètres formels pour écrire ce batch :

Exemple : affiche.bat
Echo %2
Echo %1

C :>affiche rouge noir
noir
rouge

c:\>_

Rouge et noir sont des paramètres réels et variables, %1 et %2 sont des paramètres formels. Au moment de l'exécution du fichier, le paramètre formel prend la valeur contenue dans le paramètre réel.

C :>nom-fichier-batch param1 param2 ...

Exemple :

C :>affiche rouge noir

nom-fichier-batch : 1^{er} paramètre 2^{ème} paramètre

affiche.bat

Les paramètres formels s'écrivent %n

- avec n = de 0 à 9 :
- avec n qui représente la position du paramètre dans la liste des paramètres réels fournis lors du lancement du fichier de commande.
- %0 représente le nom du fichier de commande lui-même.
- %1 représente la valeur du 1^{er} paramètre.
- %n représente la valeur du n^{ième} paramètre.

Pour exécuter un batch avec des paramètres variables, l'utilisateur doit fournir au batch des valeurs appelées paramètres réels : il passe des paramètres au batch.

Exécution d'un fichier BATCH : Nom_Fichier.BAT [P1] [P2] ... [Pn]

Conclusion :

Le passage de paramètres à un fichier batch permet de généraliser l'utilisation du fichier. Les paramètres formels servent à la définition des paramètres dans un batch. Les paramètres réels servent à l'exécution du batch.

3 – Les commandes spécifiques aux fichiers BATCH

Créer un fichier batch, c'est écrire des lignes de commandes. Quelques commandes ont été spécialement conçues pour les fichiers batch.

CALL	Appel d'un autre fichier batch
ECHO	Active ou désactive l'affichage de commande à l'écran (On/Off)
FOR	Répète une même commande plusieurs fois (FOR %% variable IN 'jeu' DO <commande>)
GOTO	Effectue un branchement vers une étiquette (nommée :ETIQ)
IF	Permet l'utilisation conditionnelle des commandes
PAUSE	Marque une pause au cours de l'exécution et permet d'afficher un message à l'écran
REM	Insère une ligne de commentaire dans le fichier, ne s'affiche pas à l'écran sauf si ECHO est sur ON.
CLS	Permet d'effacer l'écran (retour du curseur en haut à gauche).
DO	Commande permettant l'exécution d'un groupe de commandes.

A – L'instruction ECHO

MS-DOS affiche sur l'écran les commandes des fichiers batch avant de les exécuter.

L'instruction **ECHO** dirige cette visualisation.

- ECHO OFF empêche cette visualisation
- ECHO ON rétablit cette visualisation (mode normal)
- ECHO sans argument, délivre à l'écran le mode actuel
- ECHO <msg> dirige le contenu du message vers l'écran

Exemple : ECHO Début du fichier batch

Remarque :

@ N'affiche pas l'ECHO de la commande qui suit à la console. Par exemple, @**ECHO** OFF empêchera l'affichage des commandes exécutées après et y compris elle-même.

B – L'instruction REM

REM <msg> agit comme ECHO sauf que **REM** n'est pris en compte que dans le mode ECHO ON.

En fait, ECHO s'utilise pour renseigner l'opérateur et REM pour documenter le fichier de commande (c'est un commentaire).

C – L'instruction PAUSE

L'instruction **PAUSE** suspend l'exécution du fichier de commande et affiche "Presser une touche pour continuer... "

PAUSE sert à donner à l'opérateur :

- le temps de réfléchir
- la possibilité de changer une disquette
- la possibilité d'interrompre l'exécution par CTRL-C

C – Les paramètres

%1 Le signe % suivi d'une valeur numérique (à partir de 1) d'utiliser des arguments comme des variables dans une procédure.

Exemple: si MONNOM.BAT contient la ligne **echo Hello %2 %1**

Alors, lancer la commande **monnom** avec les arguments **GUETTA David**

affichera : Hello David GUETTA

D – L'instruction IF

L'instruction IF

L'instruction conditionnelle se présente sous la forme **IF [NOT] <condition> <commande>**

Tester l'égalité : = =

IF %1 == STOP GOTO FIN

Tester l'existence d'un fichier : EXIST

IF NOT EXIST C:\MRIM1\RESEAUX\COURS\%1.pas GOTO erreur

Exemple :

Créer un fichier de commande affichant un message pour dire si le fichier passé en paramètre est présent ou non dans le répertoire courant.

Exemple d'appel : trouve LesReseaux.txt

On trouvera dans le fichier trouve.bat les commandes suivantes :

IF NOT EXIST %1 GOTO erreur

ECHO Ce fichier est présent

GOTO fin

:erreur

ECHO Ce fichier est introuvable

:fin

Exemple :

C:\MRIM1A\RESEAUX\COURS>trouve LesReseaux.txt

Ce fichier est prUsent !

C:\MRIM1A\RESEAUX\COURS>trouve Les Reseaux.txt

Ce fichier est introuvable !

IF Permet de tester une condition pour effectuer des traitements conditionnels. Si la condition est VRAIE, on exécute la commande qui se trouve sur la même ligne (en général un saut), si elle est fausse on passe directement à la ligne suivante.

Les opérateurs possibles avec la commande IF :

EQU - égal à

NEQ - différent de

LSS - inférieur à

LEQ - inférieur ou égal à

GTR - supérieur à

GEQ - supérieur ou égal à

E – Les étiquettes et GOTO

:LABEL Un nom précédé de ":" constitue une étiquette (destination d'un GOTO).

Le Label et GOTO

Le label est une chaîne de 1 à 8 caractères alphanumériques. Il apparaît en position d'argument dans l'instruction GOTO. Utilisé comme étiquette sur une ligne de commande, il doit être précédé de deux points :

Exemple : **GOTO** FIN

.....

:FIN

Si le label n'est pas défini en position d'étiquette lors d'un GOTO, l'exécution du fichier est stoppée. Le message « label not found » s'affiche.

GOTO :LABEL

La procédure continue à partir de la ligne qui commence par :LABEL. Les commandes entre GOTO et la ligne de :LABEL ne sont pas exécutées. Le programme fait un "saut".

F - La structure de contrôle FOR

FOR Exécute la commande que vous spécifiez sur le groupe définit

La syntaxe est : FOR (%% variable) IN (jeu) DO <commande>

Jeu représente les noms de fichiers et doit être écrit entre parenthèse.

Exemple : Pour afficher le contenu de tous les fichiers « .txt » et « .pdf » du répertoire courant COURS on pourrait saisir .

```
FOR %1 IN (*.pdf *.txt) DO type %1.
```

Vérification de ce que l'on a dans ce répertoire :

```
C:\MRIM1\RESEAUX\COURS>dir
```

Le volume dans le lecteur C n'a pas de nom.

Le numéro de série du volume est 08EB-1312

```
Répertoire de C:\MRIM1\RESEAUX\COURS
26/11/2007 19:01 <REP>      .
26/11/2007 19:01 <REP>      ..
26/11/2007 19:00          15 LeCablage.txt
26/11/2007 19:00          18 LesReseaux.txt
26/11/2007 19:01          22 MaterielReseau.pdf
26/11/2007 19:01          15 ModeleOSI.txt
26/11/2007 19:02          22 SecuriteReseau.pdf
26/11/2007 19:01          11 TCP-IP.txt
        6 fichier(s)          103 octets
        2 Rép(s)  2 125 905 920 octets libres
```

Résultat :

```
C:\MRIM1\RESEAUX\COURS>FOR %1 IN (*.pdf *.txt) DO type %1
```

```
C:\MRIM1\RESEAUX\COURS>type MaterielReseau.pdf
MATERIEL RESEAU .PDF
```

```
C:\MRIM1\RESEAUX\COURS>type SecuriteReseau.pdf
SECURITE RESEAU .PDF
```

```
C:\MRIM1\RESEAUX\COURS>type LeCablage.txt
LE CABLAGE .TXT
C:\MRIM1\RESEAUX\COURS>type LesReseaux.txt
LES RESEAUX .TXT
```

```
C:\MRIM1\RESEAUX\COURS>type ModeleOSI.txt
MODELE OSI .TXT
C:\MRIM1\RESEAUX\COURS>type TCP-IP.txt
```

Pour optimiser l'affichage on se sert de @echo :

```
C:\MRIM1A\RESEAUX\COURS>FOR %1 IN (*.pdf *.txt) DO @echo type %1
type MaterielReseau.pdf
type SecuriteReseau.pdf
type LeCablage.txt
type LesReseaux.txt
type ModeleOSI.txt
type TCP-IP.txt
```

On pourrait aussi taper directement cette ligne de commande, plutôt que d'aller se placer dans le bon répertoire COURS.

```
C:\Documents and Settings\Thiery>cd C:\MRIM1A\RESEAUX\COURS & FOR %1 IN (*.pdf *.txt) DO @echo type %1
```

L'instruction SHIFT

Cette instruction permet d'utiliser plus de 9 paramètres réels par simple décalage.

Après chaque instruction SHIFT, le paramètre réel 0 disparaît, le paramètre réel 1 devient 0... le paramètre 10 qui ne pouvait être atteint devient 9 donc disponible et ainsi de suite.

SHIFT Change la position des paramètres positionnels (arguments) passés à une procédure.

Exemple :**Afficher le contenu des répertoires donnés en paramètre sans connaître au préalable leur nombre.**

Exemple d'appel depuis le répertoire courant : affRep \MRIM1A\RESEAUX\COURS

Contenu du fichier affRep.bat :

```
REM fichier batch affichage répertoire
ECHO OFF
:rep-suiv
IF %1 == STOP GOTO FIN
DIR /P %1
SHIFT
PAUSE
GOTO rep-suiv
:FIN
ECHO listage des répertoires terminé
ECHO ON
```

Commande :

```
C:\Documents and Settings\Thiery>affRep \MRIM1A\RESEAUX\COURS
```

Résultat :

```
C:\Documents and Settings\Thiery>REM fichier batch affichage rÚpertoire
```

```
C:\Documents and Settings\Thiery>ECHO OFF
```

Le volume dans le lecteur C n'a pas de nom.

Le numéro de série du volume est 08EB-1312

Répertoire de C:\MRIM1A\RESEAUX\COURS

```
26/11/2007 19:01 <REP> .
26/11/2007 19:01 <REP> ..
26/11/2007 19:00      15 LeCablage.txt
26/11/2007 19:00      18 LesReseaux.txt
26/11/2007 19:01      22 MaterielReseau.pdf
26/11/2007 19:01      15 ModeleOSI.txt
26/11/2007 19:02      22 SecuriteReseau.pdf
```

26/11/2007 19:01 11 TCP-IP.txt
6 fichier(s) 103 octets
2 Rép(s) 2 116 947 968 octets libres
Appuyez sur une touche pour continuer...
Le volume dans le lecteur C n'a pas de nom.
Le numéro de série du volume est 08EB-1312

Répertoire de C:\Documents and Settings\Thiery

10/12/2007 12:35 <REP> .
10/12/2007 12:35 <REP> ..
22/08/2005 13:33 <REP> .gimp-1.2
09/10/2006 19:21 0 .gtk-bookmarks
02/11/2007 16:38 0 121310
06/12/2007 22:35 <REP> Bureau
26/11/2007 18:40 0 cd
26/11/2007 12:30 0 color
21/09/2007 11:41 <REP> Favoris
27/12/2004 11:30 8 529 gsview32.ini
26/11/2007 12:30 0 md
22/11/2006 16:38 <REP> Menu Démarrer
07/12/2007 17:58 <REP> Mes documents
12/11/2007 17:23 <REP> MRIM1
26/11/2007 12:30 0 NOTEPAD
10/12/2007 12:36 5 767 168 ntuser.dat
30/11/2007 10:59 600 PUTTY.RND
26/11/2007 12:22 0 SET
02/11/2007 16:38 0 thierry
10/02/2004 18:30 <REP> WINDOWS
24/06/2004 17:40 186 566 ~
12 fichier(s) 5 962 863 octets
9 Rép(s) 2 116 947 968 octets libres
Appuyez sur une touche pour continuer...

Que pouvez- vous remarquer ???

Autres mots-clés à utiliser avec FOR :

- eol=c** - spécifie un caractère de commentaire de fin de ligne (un seul)
- skip=n** - spécifie le nombre de lignes à ignorer en début de fichier.
- delims=xxx** - spécifie un ensemble de délimiteurs. Ceci remplace l'ensemble de délimiteurs par défaut qui sont l'espace et la tabulation.
- tokens=x,y,m-n** - spécifie les parties de chaque ligne devant être transmises au corps de FOR à chaque itération. Ceci provoquera l'allocation de noms de variables supplémentaires. La forme m-n est une étendue spécifiant les parties allant de m à n. Si le dernier caractère de la chaîne tokens= est une astérisque, alors une variable supplémentaire est allouée et reçoit le texte restant dans la ligne suivant la dernière partie analysée.
- usebackq** - spécifie que la nouvelle sémantique est en place, lorsqu'une chaîne entre guillemets inversés est exécutée en tant que commande et une chaîne entre guillemets simples est une chaîne de commande littérale et permet l'utilisation de guillemets

doubles pour citer des noms de fichiers.

Exercice :

**Faites un fichier texte portant le nom de « serviceJour » puis transformer le en .bat
Pour l'exécuter.**

Il devra donner le choix entre avoir :

1 la possibilité d'avoir l'heure du moment.

2 la possibilité d'avoir la date du moment.

3 de sortir du programme.

Et ce indépendamment de l'option choisie. C'est à dire, qu'on doit pouvoir choisir de ne pas afficher l'heure et d'avoir la date et sortir, ou d'afficher la date sans afficher l'heure et sortir, ou de tout afficher.