

Stéphane Grare

Les fichiers

Batch

**Concevez rapidement des fichiers batch en utilisant
les commandes MS-Dos...**

Batch

Stéphane Grare

Un fichier Batch est un fichier qui regroupe une suite logique de commandes MS-Dos, et possédant l'extension ".bat" et est directement exécutable par le système. Les commandes sont enregistrées ligne par ligne et seront exécutées séquentiellement. La programmation Batch nécessite une connaissance minimum de l'environnement Dos. En fait, un fichier Batch contient simplement une suite de commandes que vous pourriez taper sous l'invité (prompt) du Dos, chaque nouvelle ligne du fichier correspondant à une nouvelle commande. Néanmoins, certaines commandes ne sont utilisables que dans les fichiers Batch du fait de leur inutilité dans l'environnement de commande Dos. C'est ce qui permet d'automatiser certaines tâches. Leur utilité est, par exemple, quand il faut répéter toujours la même série de commandes. À titre d'exemple, nous pourrions évoquer le changement de répertoire et peut-être aussi la commande Format qu'on fait souvent suivre de la commande CHKDSK pour vérifier si la disquette a bien été formatée.

INTRODUCTION

L'intérêt des Batches est donc d'automatiser des tâches répétitives effectuées sous DOS. Les fichiers Batch peuvent également utiliser toutes les commandes DOS, ce qui rend disponible pour le programmeur un grand nombre de fonctions. Enfin, leur taille est relativement légère par rapport à d'autres programmes, ce qui facilite leurs "transferts" sur différents disques et supports de stockage. Cependant...

- Le langage Batch n'est pas compilé, il est interprété par COMMAND.COM ce qui rend plus lent l'exécution de programmes Batch par rapport à des applications écrites directement en langage machine,
- Les fichiers Batch sont directement éditables, donc votre code n'est pas "protégé" à la copie par d'autres programmeurs,
- Enfin, et surtout, des opérations élémentaires telles que le traitement de chaînes de caractères, d'opérations mathématiques... n'existent pas sous DOS, ce qui implique l'usage de programmes externes (s'ils existent, selon les cas).

TABLE DES MATIERES

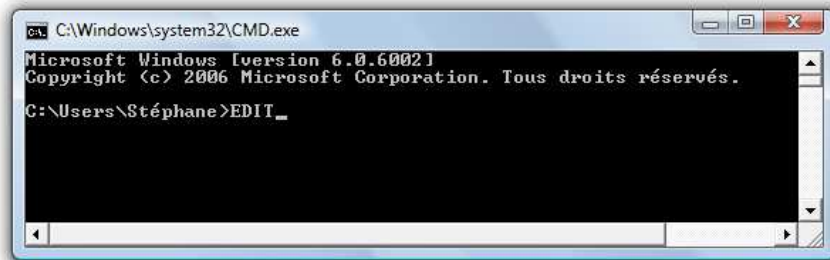
- Création d'un fichier Batch
- Notions de bases sur les fichiers Batch
- Les commandes spécifiques
- Passage de paramètres
- Les Boucles
- Aller plus loin dans les fichiers Batch
- Exemple de fichiers Batch

1 – Création d'un fichier Batch :

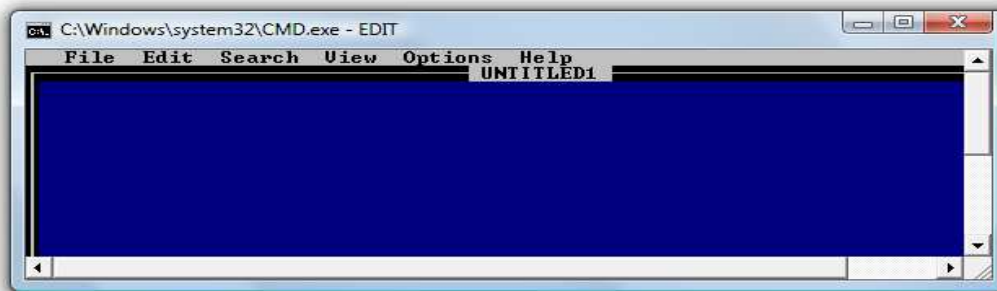
Le terme "Batch" désigne un fichier contenant une suite de commandes qui seront traitées automatiquement. Nous appelons aussi cela un "traitement par lot". Afin de créer votre premier fichier Batch, suivez cette procédure :

- 1) Ouvrez un éditeur de texte : Le Bloc-notes par exemple.
- 2) Inscrivez vos lignes de commandes.
- 3) Enregistrez votre fichier texte.
- 4) Cliquez sur le nom du fichier puis appuyez sur la touche F2. Vous serez en mode "Édition".
- 5) Renommez le fichier en changeant l'extension .txt en .bat. À la question : "Voulez-vous vraiment renommer le fichier", répondez par Oui. Attention de désactiver au préalable la case "Masquer les extensions de fichiers dont le type est connu" dans les options avancées de l'Explorateur Windows.

Pour créer un fichier Batch, vous pouvez également utiliser directement l'interpréteur de commandes : pour cela, utilisez la commande **EDIT** :



Appeler alors la commande **EDIT** pour l'éditeur de texte :



2 – Notions de base sur les fichiers Batch :

Malheureusement, la majorité des sources intéressantes qui pourront vous aider à approfondir les techniques de la programmation Batch sont souvent en anglais... "Xset" est un outil indispensable pour apprendre à écrire des fichiers Batch sous toutes les versions de Windows. De plus, Marc Stern propose de nombreux didacticiels à partir de son site Internet : <http://xset.tripod.com/>.

Jerold Schulman qui m'a autorisé à citer quelques-uns de ces scripts anime un site Internet qui propose d'innombrables pages sur la programmation Batch : <http://www.jsiinc.com/>. C'est, par ailleurs, un des meilleurs sites traitant des systèmes NT de Windows.

D'excellents scripts sont également visibles à partir de cette adresse : www.uwasa.fi/~ts/http/http2.html#batch. Timo Salmi propose une multitude de solutions permettant de résoudre les problèmes les plus courants quand on commence à se lancer dans la programmation Batch.

Ritchie Lawrence met en lignes de nombreux modèles de scripts et propose quelques utilitaires fort bien faits qui vous permettront de réaliser des choses étonnantes en mode Ms-Dos ou en Invite de commandes. L'adresse de son site est la suivante : <http://www.commandline.co.uk/index.html>.

Enfin, une source inépuisable de trouvailles est proposée à partir de ce forum recouvrant une multitude de thématiques dont Ms-Dos et l'Invite de commandes : http://www.experts-exchange.com/Operating_Systems/MSDOS. La plupart des exemples de scripts expliqués dans ce chapitre s'inspirent directement ou indirectement des techniques présentées par ces différents auteurs.

Quelle différence entre l'extension .cmd .btm et .bat ?

Un fichier **.cmd** ou **.btm** ne sera pas reconnu en tant que tel par Windows 9X. Nous pouvons voir cela comme une sorte de

garde-fou si tel fichier de commande est parfaitement incompatible avec ces versions de Windows. Par ailleurs, le processus est légèrement différent : dans le cas d'un fichier **.bat** ou **.cmd**, chaque ligne du fichier est exécutée individuellement et le fichier fermé puis ouvert à chaque lecture d'une nouvelle commande. Dans le cas d'un fichier **.btm**, le fichier n'est ouvert qu'une fois, puis lu en mémoire et enfin fermé. C'est donc a priori le mode le plus rapide. Surtout si ce sont des commandes internes qui sont exécutées.

Se servir des parenthèses dans les fichiers de scripts :

Si vous souhaitez rediriger le résultat de différentes commandes dans un même fichier texte. Par exemple :

```
@echo off
DIR /s /b *.doc >> resultat.txt
DIR /b *.dot >> resultat.txt
REM etc.
@echo on
```

Il est dans ce cas plus simple de saisir :

```
@echo off
(
DIR /s /b *.doc
DIR /s /b *.dot
) >> resultat.txt
@echo on
```

Cela vous évitera de spécifier plusieurs fois le même fichier de sortie. Par ailleurs, c'est une manière de créer de toutes pièces un fichier texte :

```
@echo off
(
@echo Bonjour,
@echo Tout le monde !
) > test.txt
```

Dans ce dernier cas, les commandes sont regroupées.

Afficher le code de sortie :

À chaque fois que vous saisissez une commande, cette dernière produit un code de sortie qui est principalement :

1 : La commande a renvoyé une erreur
0 : La commande n'a pas renvoyé d'erreur.

C'est la variable **%Errorlevel%** qui est chargée de suivre les fluctuations de la réussite ou non des commandes exécutées. La commande "Set" teste les "Errorlevel" en partant de la plus petite valeur puis en procédant par incrémentation de 1. La commande "**GOTO**" démarre de la plus grande valeur puis procède par décrémentation de 1.

Gérer les caractères accentués :

Les différences entre la norme Ansi (Windows) et OEM (Ms-Dos) font que si vous vous servez d'un éditeur de texte comme le Bloc-Notes les caractères accentués ne s'afficheront pas correctement dans les sorties écran en mode "Console". Ainsi, la phrase "Je répète : "Vous êtes un entêté" s'affichera comme cela : Je rÚpPte : "Vous Ôtes un entÔtÚ". Il y a différentes solutions :

Vous pouvez toujours vous servir de l'éditeur de texte prévu dans toutes les versions de Windows : "**Edit**".

Il existe de nombreux programmes vous permettant d'opérer la conversion Ansi vers OEM et vice-versa. "OEMANSI.exe" est un programme écrit en français que vous pouvez télécharger à partir de cette adresse :

<http://perso.wanadoo.fr/andre.araste/tele2.htm>

Décompressez l'archive ZIP nommée Oemansi.zip puis double-cliquez sur ce fichier exécutable : Setup.exe afin de lancer l'installation de ce logiciel. Imaginons que notre fichier à convertir s'appelle Test.bat suivez la procédure suivante :

- 1) Cliquez sur Démarrer/Tous les programmes/OEMANSI/OEMANSI.
- 2) Cliquez sur le bouton Ouvrir Fichier.
- 3) Dans la liste déroulante Fichiers de type :, sélectionnez Tous les fichiers (*.*) puis votre fichier Batch.
- 4) Cochez le bouton radio ANSI(Win) puis la flèche dirigée vers le bas.

Il est possible d'effectuer l'opération inverse !

5) Cliquez sur le bouton Sauvegarde OEM puis lancer le processus de conversion.

6) Confirmez le remplacement ou non de l'ancienne version.

Gérer les caractères réservés :

Ces caractères ne peuvent pas être employés "tels quels" dans un script. Le signe % pour être "compris" dans un fichier de script doit être redoublé. Par ailleurs, nous avons déjà vu que pour afficher un caractère réservé il faut le faire précéder du signe ^. Dans ce cas-là, il ne sera pas assimilé à une commande, mais bien à un caractère. Afin d'afficher les signes : ! ^ & < > >> " | créez un fichier Batch contenant cette commande :

```
@echo off
ECHO ^! ^% ^^ ^& ^< ^> ^>^> ^" ^|
@echo on
```

Une manière d'afficher les caractères de redirection :

```
@echo off
<nul (SET /p z=Le^|texte^|de^|sortie) >sortie.txt
@echo on
```

Nous redirigeons le produit de la commande "**SET**" vers le fichier Sortie.txt, mais en désactivant toute sortie-écran. L'utilisation de la commande "**SET**" sera expliquée plus loin.

3 – Liste des commandes :

Dans la rubrique suivante, nous détaillerons certaines commandes spécifiques. Il s'agit ici d'un aperçu rapide des commandes existantes en batch.

ANSI.SYS	Définit les fonctions qui modifient l'affichage, contrôlent le déplacement du curseur et réaffectent les touches.
APPEND	Permet aux programmes d'ouvrir les fichiers de données qui se trouvent dans les répertoires spécifiques, comme s'ils figuraient dans le répertoire en cours.
ARP	Affiche, ajoute, et supprime les informations arp des dispositifs du réseau.
ASSIGN	Permet d'assigner une nouvelle lettre à un lecteur
ASSOC	Affiche les associations de fichier.
AT	Programme une heure pour exécuter une commande.
ATMADM	Liste les connexions et adresses vu par Windows ATM call manager.
ATTRIB	Affiche et change les attributs de fichiers.
BATCH	Fichier qui exécute une série de commande.
BREAK	Active / désactive CTRL + C dispositif
CACLS	Affiche et modifie le fichier ACL.
CALL	Appelle un fichier batch à partir d'un autre fichier batch.
CD	Change de répertoire.
CHCP	Permet de changer le jeu de caractères.
CHDIR	Idem que CD.
CHKDSK	Vérifie que le disque dur en format FAT ne contient pas d'erreurs.
CHKNTFS	Vérifie que le disque dur en format NTFS ne contient pas d'erreurs.
CHOICE	La commande choice demande à l'utilisateur de saisir une des lettres proposées. La récupération du choix se fait par la commande errorlevel.

CLS	Efface l'écran.
CMD	Ouvre la fenêtre de commande DOS.
COLOR	Change les couleurs de l'arrière-plan et de la police de la fenêtre
COMMAND	Idem que CMD
COMP	Compare des fichiers.
COMPACT	Comprime et décomprime des fichiers
CONTROL	Ouvrez les icônes de panneau de commande de la fenêtre DOS.
CONVERT	Converti du format FAT vers NTFS.
COPY	Copie un ou plusieurs fichiers vers une différente destination.
CTTY	Modifie les périphériques et d'entrées standards
DATE	Affiche ou modifie la date système.
DEBUG	Utilitaire permettant des programmes en assembleur afin de modifier les paramètres Hardware. Debug utility to create assembly programs to modify hardware settings.
DEFRAG	Permet de défragmenter un disque dur.
DEL	Supprime un ou plusieurs fichiers.
DELETE	Idem que DEL.
DELTREE	Efface un ou plusieurs fichiers et/ou répertoires
DIR	Affiche la liste des fichiers et des sous-répertoires d'un répertoire.
DISABLE	Désactive les services ou drivers windows.
DISKCOMP	Compare les contenus de deux disquettes.
DISKCOPY	Copie le contenu d'une disquette sur une autre.
DOSKEY	Modifie les lignes de commande, rappelle des commandes Windows, et permet de créer des macros.
DOSSHELL	Représente les répertoires et les fichiers sous forme d'icônes dans un environnement de menus déroulants qui contiennent les principales commandes du système d'exploitation.
DRIVPARM	Redéfinit les paramètres d'un lecteur.
ECHO	Affiche des messages... l'écran ou active/désactive l'affichage des commandes.
EDIT	Affiche et édite des fichiers.
EDLIN	Idem que EDIT.
EMM386	Charge extended Memory Manager.
ENABLE	Désactive les services ou drivers windows
ENDLOCAL	Stoppe la localisation des modifications de l'environnement dans un fichier de commandes.
ERASE	Supprime un ou plusieurs fichiers.
EXIT	Quitte l'interpréteur de commandes (CMD.EXE).
EXPAND	Décomcodesse un fichier.
EXTRACT	Utilitaire de décomcodession des fichiers archives cab de windows 95.
FASTHELP	Affiche des informations sur les commandes de Windows.

FC	Compare deux fichiers ou groupes de fichiers, et affiche les différences entre eux..
FDISK	Configurer / partitionner un disque dur.
FIND	Cherche une chaîne de caractères dans un ou plusieurs fichiers.
FINDSTR	Cherche des chaînes de caractères dans un ou plusieurs fichiers.
FIXBOOT	Ecrit un nouveau secteur de boot.
FIXMBR	Ecrit un nouveau secteur de boot sur un lecteur..
FOR	Exécute une commande sur chaque fichier d'un groupe de fichiers.
FORMAT	Formate un disque pour utilisation avec Windows.
FTP	Commande pour se connecter et opérer sur un serveur FTP.
FTYPE	Affiche ou modifie les types de fichiers utilisés dans les associations d'extensions.
GOTO	Poursuit l'exécution d'un fichier de commandes à une ligne identifiée par une étiquette.
GRAFTABL	Permet à Windows d'afficher un jeu de caractères en mode graphique..
HELP	Affiche des informations sur les commandes de Windows.
IF	Effectue un traitement conditionnel dans un fichier de commandes.
IFSHLP.SYS	Gestionnaire de fichier 32-bit.
IPCONFIG	Commande réseau pour voir / configurer les paramètres réseau.
KEYB	Charger un pilote de clavier.
LABEL	Crée modifie ou supprime le nom de volume d'un disque.
LH	Charge un programme en zone de mémoire supérieure, ce qui libère de la mémoire conventionnelle pour d'autres programmes.
LISTSVC	Affiche tous les services, pilotes et types de démarrage.
LOADFIX	Charge un programme sur les codemiers 64k.
LOADHIGH	Charge un programme résident dans la mémoire supérieure.
LOCK	Verrouille l'accès à un disque par une application
LOGON	Cette commande affiche les installations de Windows et de Windows NT détectées, et demande le mot de passe de l'administrateur local pour la copie de Windows à laquelle vous souhaitez vous connecter. NB: Au bout de 3 échecs, la console s'arrête et l'ordinateur redémarre...
MAP	Affiche le nom d'un lecteur.
MD	Crée un répertoire.
MEM	Affiche la mémoire du système.
MKDIR	Crée un répertoire.
MODE	Configure un périphérique du système.
MORE	Affiche la sortie écran par écran.
MOVE	Déplace un ou plusieurs fichiers d'un répertoire à un autre.
MSAV	Microsoft anti-virus.
MSD	Utilitaire de Diagnostics.
MSCDEX	Utilitaire permettant d'utiliser, de charger et accéder à un CD-ROM.

NBTSTAT	Mise à jour du cache du fichier Lmhosts
NET	Mettre à jour, réparer, ou voir le réseau et les paramètres réseau.
NETSH	Configure les informations d'un réseau dynamique et statique.
NETSTAT	Afficher les statistiques de protocole et l'état actuel des connexions NetBIOS sur TCP/IP
NLSFUNC	Charge un jeu de caractère spécifique.
NSLOOKUP	Permet de tester un serveur DNS.
PATH	Affiche ou définit le chemin de recherche des fichiers exécutables.
PATHPING	Outil de trace qui combine les caractéristiques du ping et de tracert avec des informations additionnelles qu'aucune de ces 2 commandes ne fournit.
PAUSE	Interrompt l'exécution d'un fichier de commandes et affiche un message
PING	Affiche les informations de connexion à une autre machine.
POPD	Restaure la valeur précédente du répertoire courant enregistré par PUSHHD.
POWER	Conservez la puissance d'un ordinateur portable.
PRINT	Imprime un fichier texte.
PROMPT	Modifie l'invite de commande de Windows.
PUSHHD	Enregistre le répertoire courant puis le modifie.
QBASIC	Ouvre QBASIC.
RD	Supprime un répertoire vide.
REN	Renomme un ou plusieurs fichiers.
RENAME	Renomme un ou plusieurs fichiers.
RMDIR	Supprime un répertoire vide.
ROUTE	Manipule les tables de routage du réseau
RUNAS	Autorise un utilisateur à exécuter un programme un autre ordinateur.
SCANDISK	Lance l'utilitaire scandisk.
SCANREG	Scanne et restaure la base de registre.
SET	Affiche, définit ou supprime des variables d'environnement Windows.
SETLOCAL	Commence la localisation des changements de l'environnement dans un fichier de commandes.
SETVER	Définit le numéro de version que MS-DOS fournit à un programme.
SHARE	Permet de verrouiller les fichiers lorsqu'ils peuvent être accédés par plusieurs programmes en même temps
SHIFT	Modifie la position des paramètres remplaçables dans un fichier de commandes.
SHUTDOWN	Éteint l'ordinateur.
SMARTDRV	Crée un cache disque dans la mémoire étendue.
SORT	Trie les éléments en entrée.
START	Lance une fenêtre pour l'exécution du programme ou de la commande.
SUBST	Affecte une lettre de lecteur à un chemin d'accès.
SWITCHES	Configure le clavier étendu.

SYS	Transférer les fichiers système vers un lecteur.
TELNET	Telnet vers un autre ordinateur / système.
TIME	Affiche ou définit l'heure de l'horloge interne du système.
TITLE	Définit le titre de la fenêtre MS-DOS.
TRACERT	Permet de voir le chemin parcouru entre votre poste et l'ordinateur qui héberge le site
TREE	Représente graphiquement l'arborescence d'un lecteur ou d'un chemin.
TYPE	Affiche le contenu d'un fichier texte.
UNDELETE	Récupère un fichier qui a été effacé.
UNFORMAT	Annule un formatage.
UNLOCK	Unlock un disque dur.
VER	Affiche le numéro de version.
VERIFY	Indique à Windows s'il doit ou non vérifier que les fichiers sont écrits correctement sur un disque donné.
VOL	Affiche le nom et le numéro de série du volume.
XCOPY	Copie des fichiers et des arborescences de répertoires.

Pour de plus ample détail sur chacune de ses commandes, consulter l'aide en ligne de commande. Pour cela, il suffit d'écran la commande exemple : DEL /?

```

C:\Windows\system32\CMD.exe
Microsoft Windows [version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. Tous droits réservés.

C:\Users\Stéphane>DEL /?

```

Et l'aide s'affiche :

```

C:\Windows\system32\CMD.exe
Microsoft Windows [version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. Tous droits réservés.

C:\Users\Stéphane>DEL /?
Supprime un ou plusieurs fichiers.

DEL [/P] [/F] [/S] [/Q] [/A[:attributs]] noms
ERASE [/P] [/F] [/S] [/Q] [/A[:attributs]] noms

noms          Spécifie une liste d'un ou plusieurs fichiers ou répertoires.
               Les caractères génériques peuvent être utilisés pour
               supprimer plusieurs fichiers. Si un répertoire est spécifié,
               tous les fichiers qu'il contient seront supprimés.

/P           Demande une confirmation avant de supprimer un fichier.
/F           Force la suppression de fichiers en lecture seule.
/S           Supprime les fichiers dans tous les sous-répertoires.
/Q           Pas de confirmation avant suppression avec car. génériques.
/A          Suppression en fonction des attributs
attributs    R Fichiers en lecture seule      S Fichiers système
             H Fichiers cachés                A Fichiers prêts pour archive
             I Fichiers indexés sans contenu  L Points d'analyse
             - Préfixe de négation

Si les extensions de commandes sont activées, DEL et ERASE sont modifiées
comme suit :
La logique d'affichage du commutateur /S est inversée. Elle
Appuyez sur une touche pour continuer...

```

4 – Les commandes spécifiques :

ECHO ON/OFF

Permet d'activer / désactiver l'affichage des commandes qui sont exécutées. Par défaut, c'est la commande **ECHO ON** qui est utilisée. L'utilisation de l'arobase permet de biffer le statut de la commande "**ECHO**". Par ailleurs, la commande "Echo off" évite l'affichage des commandes contenues dans le script.

```
@echo off
ECHO Bonjour
@echo on
```

ECHO message

Permet l'affichage du message. MS-DOS n'est pas sensible pour les commandes à la différence entre les majuscules et les minuscules, que vous écriviez echo ou Echo, ou bien encore ECHO ou Echo, le résultat sera le même. N'utilisez pas d'accent, car MS-DOS va remplacer les caractères accentués par des symboles.

```
J'ai útú rePue ó mon examen !!!
```

ECHO.

Affiche une ligne blanche. Tilmo Salmi sur cette page web : www.uwasa.fi/~ts/http/http2.html#batch signale un problème concernant l'utilisation de la commande "**ECHO**" quand cela s'applique aux variables. Comparez la sortie-écran de ces deux fichiers Batch :

```
@echo off
SET variable=Ceci est un test
ECHO %variable%
SET variable=
ECHO %variable%
@echo on
```

Seul le second fichier Batch n'affiche pas d'erreur.

```
@echo off
SET variable=Ceci est un test
ECHO. %variable%
set variable=
ECHO. %variable%
@echo on
```

L'explication est simple : la commande "**ECHO**" suivie d'un point force l'affichage d'une ligne vide.

REM

Permet d'insérer des commentaires dans le fichier Batch. Les commentaires sont très importants dans les programmes.

```
@echo off
REM ceci est un commentaire dans un fichier .BAT
@echo on
```

La commande **REM** vous permet d'insérer des commentaires ou de désactiver temporairement des commandes incluses dans votre fichier batch. Ce n'est pas tout à fait vrai. La commande **REM ECHO quelque chose > Sortie.txt** ne fait qu'envoyer rien au fichier Sortie.txt. La commande est donc exécutée, mais en mode "désactivé". Une sorte de coup à blanc... Pour les amoureux de la performance il est donc plus judicieux d'utiliser **::** plutôt que **REM**. Les deux points étant considérés comme l'indication d'une étiquette, la commande ne sera pas exécutée. À titre de test, saisissez tour à tour ces deux commandes :

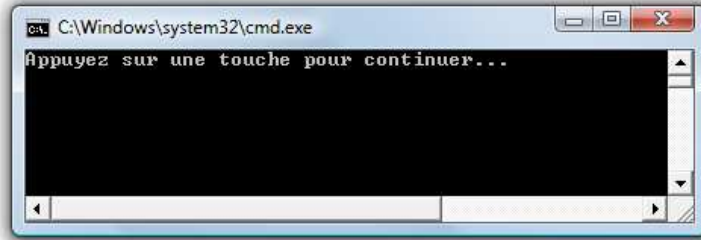
```
@echo off
REM ECHO A
:: ECHO B
@echo on
```

La seconde commande provoque un retour chariot et non un saut de ligne. Dans un fichier Config.sys il est possible de

désactiver une commande en utilisant un point-virgule : `;connexion=c:\dos\ramdrive.sys`

PAUSE

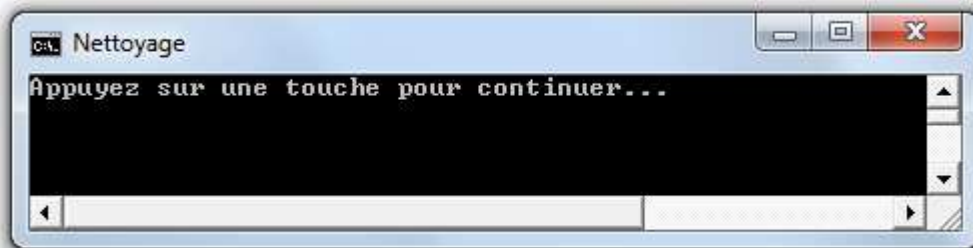
Permet de suspendre l'exécution du fichier. Un message du style « Appuyer sur une touche pour continuer » s'affiche.



TITLE

Cette commande comme son nom l'indique change le titre par défaut : c'est-à-dire `cmd.exe`, par celui que vous définissez de cette façon : `title` suivi du titre que vous avez choisi

```
@echo off
TITLE Nettoyage
@echo on
```



COLOR

Celle-ci change la couleur de la police et celle de l'arrière-plan. Après `color` il faut indiquer une couleur en hexadécimal. Voici toutes les couleurs :

0 = Noir	8 = Gris
1 = Bleu foncé	9 = Bleu clair
2 = Vert	A = Vert clair
3 = Bleu gris	B = Cyan
4 = Marron	C = Rouge
5 = Pourpre	D = Rose
6 = Kaki	E = Jaune
7 = Gris clair	F = Blanc

Le premier caractère correspond à l'arrière-plan et le second à la police.

Par exemple :

```
@echo off
COLOR C5
ECHO TEST
PAUSE
@echo on
```



Ou

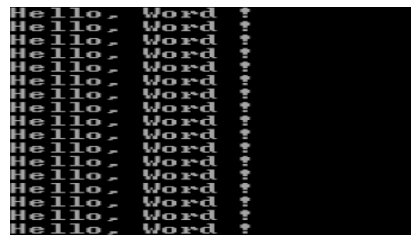
```
@echo off
COLOR 8F
ECHO TEST
PAUSE
@echo on
```



GOTO

Il s'agit d'un saut inconditionnel qui permet de casser la séquence d'exécution des commandes, de continuer l'exécution à un endroit donné différent dans le code. La commande "**GOTO**" permet d'aller au label du même nom. La marque ":Label" est un "mot" précédé de deux points (":"), correspondant à un repère utilisé par la commande **GOTO**. Un saut inconditionnel peut aussi être utilisé pour sauter des morceaux de code. Un saut inconditionnel peut être utilisé avec la commande "**IF**" pour exécuter ou ne pas exécuter du code en fonction d'une condition.

```
@echo off
:Start
echo Hello, Word !
Goto Start
```



Appuyer sur la combinaison des touches Ctrl + C : permet d'arrêter le programme en cours d'exécution.

La commande "**GOTO**" vous permet donc d'atteindre un point précis de votre script. La commande spécifiée doit porter un titre que nous appelons une étiquette. Cette ligne de titre commence obligatoirement par deux points. Nous sommes obligés d'anticiper quelque peu sur l'explication des conditions, car c'est une des utilisations privilégiées des étiquettes. Dans un nouveau fichier Batch copiez ce contenu :

```
@echo off
DIR *.doc
IF NOT errorlevel 1 GOTO Fin ELSE IF GOTO Avertissement
:Avertissement
ECHO Aucun fichier .doc !
:Fin
```

La variable %Errolevel% nous permet de saisir si la commande "DIR" a renvoyé une valeur 1 ("Échec") ou 0 ("Opération réussie"). Si au moins un fichier .doc est trouvé, nous allons directement à l'étiquette :Fin. Ou, plus exactement si la sortie d'erreur n'est pas 1 ("Échec"), alors il faut se rendre à l'étiquette :Fin. Sinon (ELSE IF) nous nous rendrons à l'étiquette :Avertissement. Si nous rajoutons une commande à la suite de l'étiquette :Fin, cette dernière sera automatiquement exécutée même si la condition n'est pas remplie. Afin d'éviter cela, il nous faut nous servir d'une étiquette spéciale : goto:eof. Notre fichier script devient alors :

```
@echo off
DIR*.doc
IF NOT errorlevel 1 goto Fin ELSE IF GOTO Avertissement
:Avertissement
ECHO Aucun fichier .doc ! & goto:eof
:Fin
ECHO Processus finit !
@echo on
```

Dans ce dernier cas, la commande goto:eof nous permet d'effectuer une césure dans le script. Cela revient à dire : "Affiche le message 'Aucun fichier .doc !' puis ne fait plus rien". En termes savants, puisque nous ne définissons pas d'étiquette, nous transférons le contrôle à la fin du script en cours.

IF

Permet d'exécuter une commande si une condition est vérifiée. **IF** n'accepte qu'une seule commande à sa droite, c'est pour cela que la commande "**GOTO**" sera régulièrement utilisée, pour exécuter ou non certaines parties du Batch. Il y a différentes formes du **IF** : IF, IF EXIST, IF ERRORLEVEL et IF NOT qui peuvent être combinés. On retrouve les syntaxes suivantes :

if [not] errorlevel Nombre Commande [else Expression]

if [not] Chaîne1==Chaîne2 Commande [else Expression]

if [not] exist NomFichier Commande [else Expression]

Si les extensions de commandes sont activées, utilisez la syntaxe suivante :

if [/i] Chaîne1 OpComparaison Chaîne2 Commande [else Expression]

if cmdextversion Nombre Commande [else Expression]

if defined Variable Commande [else Expression]

D'où :

not	Spécifie que la commande doit être exécutée uniquement si la condition est fausse.
errorlevel Nombre	Spécifie que la condition est vraie uniquement si le programme précédent exécuté par Cmd.exe a renvoyé un code de sortie égal ou supérieur à <i>Nombre</i> .
Commande	Spécifie la commande qui doit être exécutée si la condition précédente est remplie.
Chaîne1==Chaîne2	Ne reconnaît que la condition est vraie que si <i>Chaîne1</i> et <i>Chaîne2</i> sont identiques. Ces valeurs peuvent être des chaînes littérales ou des variables de fichier de commandes (par exemple, %1). Vous n'avez pas besoin d'utiliser de guillemets autour des chaînes littérales.
exist NomFichier	Ne reconnaît la condition comme étant vraie que si <i>NomFichier</i> existe.
/i	Force les comparaisons de chaînes à ne pas tenir compte de la casse. Vous pouvez utiliser /i sur la forme <i>Chaîne1==Chaîne2</i> de if . Ces comparaisons sont génériques, en ce sens que si <i>Chaîne1</i> et <i>Chaîne2</i> sont tous deux constitués uniquement de chiffres numériques, les chaînes sont converties en nombres et une comparaison numérique est effectuée.
cmdextversion Nombre	Spécifie qu'une condition est vraie uniquement si le numéro de version interne associé à l'extension de commande de Cmd.exe est supérieur ou égal à <i>Nombre</i> . La première version est égale à 1. Elle est incrémentée d'une unité lorsque des améliorations significatives sont ajoutées aux extensions de commandes. L'expression conditionnelle cmdextversion n'est jamais vraie lorsque les extensions de commandes sont désactivées (par défaut, les extensions de commandes sont activées).
defined Variable	Ne reconnaît la condition comme étant vraie que si <i>Variable</i> est défini.
Expression	Spécifie qu'une commande de ligne de commande et tous ses paramètres doivent être transmis à la commande dans une clause else .
OpComparaison	Spécifie un opérateur de comparaison en trois lettres. Le tableau suivant répertorie les valeurs valides de <i>OpComparaison</i> .

Si les extensions de commandes sont activées, il est possible d'utiliser la syntaxe suivante :

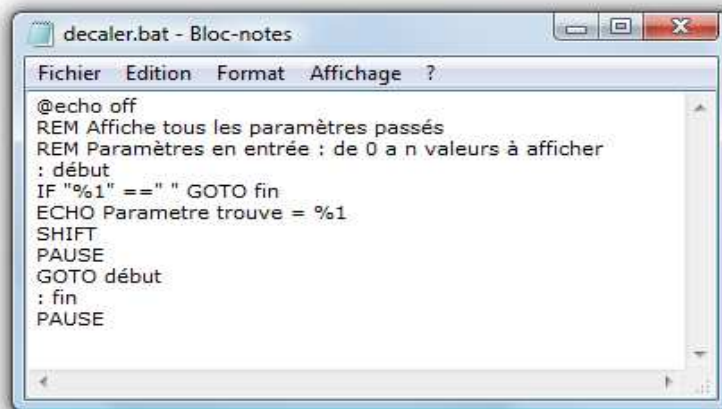
Supérieur ou égal à [/i] Chaîne1 Options_Comparaison Chaîne2 Commande [ELSE Expression]

Le commutateur /i permet de ne pas tenir compte de la casse. Tableau du récapitulatif les valeurs possibles pour les options de comparaison :

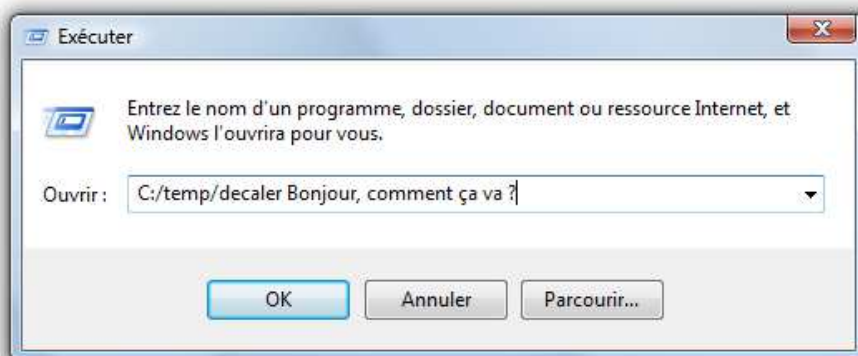
Opérateur	Description
EQU	Égal à
NEQ	Différent de
LSS	Inférieur à
LEQ	Inférieur ou égal à
GTR	Supérieur à
GEQ	Supérieur ou égal à

Prenons l'exemple suivant. On va afficher la liste des paramètres passés au programme Batch. Je crée un fichier que je nomme **decaler.bat** dans le répertoire C:\Temp.

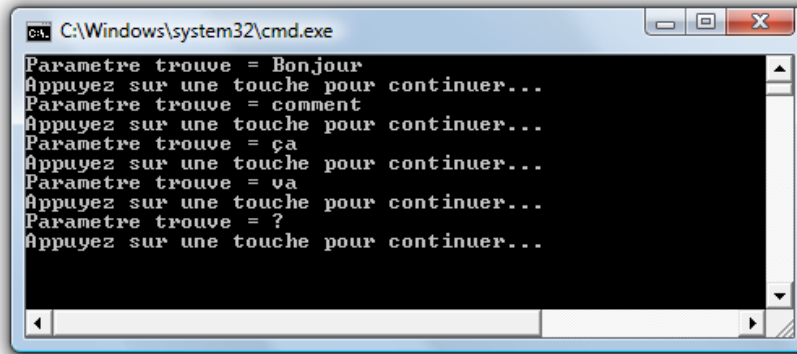
```
@echo off
REM Affiche tous les paramètres passés
REM Paramètres en entrée : de 0 a n valeurs à afficher
: début
IF "%1" == " " GOTO fin
ECHO Parametre trouve = %1
SHIFT
PAUSE
GOTO début
: fin
PAUSE
```



On va le lancer de la manière suivante, à l'aide de la commande exécutée de Windows :



À chaque valeur reçue, on clique sur une touche pour obtenir les valeurs une à une :



Examinons cette commande :

IF cmdextversion Nombre Commande [else Expression]

Le mot clé cmdextversion permet de vérifier si les extensions de commandes sont activées ou non. La première version est égale à 1. Elle doit être indiquée dans "Nombre". Voici une autre possibilité :

IF defined Variable Commande [ELSE Expression]

Ne reconnaît la condition comme étant vraie que si "Variable" est défini. Un cas particulier concerne la vérification des répertoires. Si nous souhaitons nous assurer que le répertoire Test existe bien nous devrons saisir :

IF EXIST Test\nul (@echo "Le répertoire existe bien !") ELSE md Test

Cela revient à dire que si le répertoire Test existe bien un périphérique fictif (nul) existe aussi. Nous pouvons imaginer la création d'un fichier de commande qui se chargera de vérifier si un répertoire Test existe sur le disque dur. Si la commande précédente échoue (errorlevel 0) alors il sera créé. Copiez dans un nouveau fichier Batch ces trois lignes :

```
@echo off
DIR /s Test
IF errorlevel 0 (md Test)
```

Nous pourrions écrire la dernière ligne de cette façon :

IF %errorlevel% leq 0 (md C:\Test)

Dans ce dernier cas nous nous servons de errorlevel comme d'une variable qui sera égale ou non à 0. Nous pouvons aussi saisir ce type de commande :

IF NOT EXIST C:\Test\nul md C:\Test

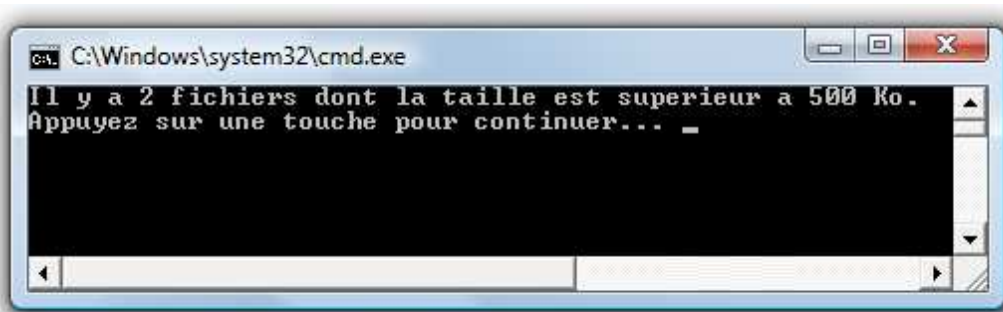
Vous pouvez conditionner la recherche à la condition que l'utilisateur saisisse le nom du fichier à rechercher :

```
@echo off
IF not exist %1 (echo Le fichier %1 est introuvable dans ce répertoire.) else echo Fichier trouvé !
```

La variable %1 représente dans ce cas la chaîne de caractères (le nom du fichier) que l'utilisateur aura saisie à la suite du nom du fichier de commande.

Prenons l'exemple suivant. On recherche dans le répertoire courant, tous les fichiers du type ***.jpg** qui ont une taille supérieure à 500 Ko (soit 500000 octets)

```
@echo off
REM Pour toutes les extensions mettre *.*
REM Ici on indique l'extension *.jpg
SET filtre=*.jpg
REM La taille est exprimée en octet
SET taille=500000
SET compteur=0
REM J'utilise l'option de comparaison /i GTR
FOR /r %%%a in (%%filtre%) DO IF /i %%%~za GTR %%taille% (
SET /a compteur+=1
)
ECHO Il y a %%compteur% fichiers dont la taille est supérieure a 500 Ko.
PAUSE
@echo on
```



Pour lister les fichiers *.jpg supérieurs à 500 Ko on modifiera le code précédent de la manière suivante :

```
@echo off
REM Pour toutes les extensions mettre *.*
REM Ici on indique l'extension *.jpg
SET filtre=*.jpg
REM La taille est exprimée en octet
SET taille=500000
SET compteur=0
ECHO Listing des fichiers *.jpg supérieur a 500 Ko.
ECHO.
REM J'utilise l'option de comparaison /i GTR
FOR /r %%a in (%%filtre%) DO IF /i %%~za GTR %%taille% (
SET /a compteur+=1
ECHO %%a
ECHO.
)
ECHO Il y a %%compteur% fichiers dont la taille est supérieure a 500 Ko.
ECHO.
PAUSE
@echo on
```

Le signe == sert uniquement à tester une égalité.

SET

Permet d'afficher, de définir ou de supprimer des variables d'environnement. Utilisée sans paramètres, la commande **SET** affiche les paramètres d'environnement en cours.

variable = valeur : Permet d'affecter une valeur à une variable

Dans l'exemple suivant, on définit la variable fichier, on lui affecte une valeur (ici "C:\Temp\Test.xls") est on affiche sa valeur à l'écran.

```
@echo off
SET fichier="C:\Temp\Test.xls"
ECHO %fichier%
PAUSE
```

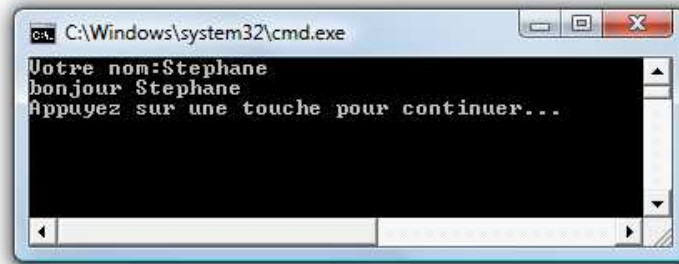


Dans l'exemple suivant, on demande à l'utilisateur de saisir son nom et on récupère l'information dans la variable :

```
@echo off
SET /p nom=Votre nom:
REM On récupère le nom de l'utilisateur saisi dans la variable "nom"
```



```
ECHO bonjour %nom%
PAUSE
@echo on
```



À noter les sous-commandes suivantes :

/a	Permet d'indiquer pour le paramètre <i>chaîne</i> une expression numérique qui est évaluée.
/p	Affecte au paramètre <i>variable</i> la valeur d'une ligne d'entrées.

Utilisation de l'option **/a** : Le tableau suivant présente les opérateurs autorisés avec l'option **/a** par ordre de priorité décroissante.

Opérateur	Opération effectuée
< >	Groupement
* / % + -	Calcul
<< >>	Décalage logique
&	ET au niveau du bit
^	OU exclusif au niveau du bit
	OU au niveau du bit
= *= /= %= += -= &= ^= = <<= >>=	Attribution
,	Séparateur d'expression

EXIST/NOT EXIST nom de fichier

Permet de vérifier l'existence ou non d'un fichier. On test la présence du fichier « Test.xls » dans le répertoire courant, c'est-à-dire dans le répertoire où on exécute le fichier batch :

```
@echo off
IF EXIST Test.xls GOTO suite
REM S'il n'existe pas, on l'indique, s'il existe on passe à la suite
ECHO N'EXISTE PAS
PAUSE
: suite
REM Action de votre choix...
PAUSE
@echo on
```

Si le fichier « Test.xls » ne se trouve pas dans le répertoire courant :



Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

