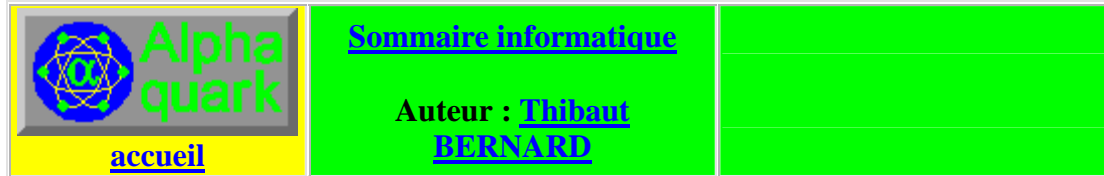


# Ms Dos

## MicroSoft Disk Operating System



### Les commandes

[Sélection d'un ou plusieurs fichiers](#)  
[Les attributs de fichiers](#)  
[Gestion de dossiers](#)  
[Gestion des fichiers](#)  
[Catalogue \(listage des noms de fichiers stockés sur mémoire de masse\)](#)  
[Visualisation du contenu d'un fichier texte](#)  
[Impression](#)  
[Lecteurs virtuels](#)  
[Divers](#)

### Les batch

[Instruction echo](#)  
[Commande choice](#)  
[Les paramètres](#)  
[Commande shift](#)  
[Divers](#)

### *Les commandes*

Pour pouvoir gérer les fichiers sur les mémoires de masse, l'ordinateur a besoin d'un logiciel appelé : système d'exploitation dont voici les principales commandes valables sur tout les ordinateurs compatible PC.

Chaque commande doit être validée par la touche 'Entrée' ('enter' ou 'Retour Chariot' en fonction des claviers).

---

## Sélection d'un ou plusieurs fichiers

Toutes les commandes suivies d'un espace puis des différentes syntaxes suivantes, agiront sur les fichiers sélectionnés :

NOM.EXT Sélection d'un fichier dont on précise le NOM et l'EXTension.

\*.EXT Sélection de tous les fichiers dont on précise l'EXTension.

B\*.\* Sélection de tous les fichiers dont le nom commence par B.

C\*.EXT Sélection de tous les fichiers dont le nom commence par C et dont on précise l'EXTension.

\*.\* Sélection de tous les fichiers sans exception.

---

## Les attributs de fichiers

Les quatre attributs qui nous intéressent ici sont : Lecture seule, archive, caché, et système. Nous allons voir les trois premiers.

Syntaxe :

ATTRIB toto.doc +r. Active la lecture seule au fichier toto.doc. On ne peut plus effacer ce fichier avec la commande erase.

ATTRIB toto.doc -r. Désactive la lecture seule au fichier toto.doc.

ATTRIB toto.doc. Affiche les attributs du fichier toto.doc.

ATTRIB \*.\*. Affiche les attributs de tous les fichiers.

ATTRIB \*.\* +r /s. Met tous les fichiers en lecture seule y compris les fichiers situés dans les sous répertoires (paramètre /s).

Paramètres : Lecture seule +r ou -r, archive +a ou -a, caché +h ou -h, et système +s ou -s.

L'attribut "lecture seule" permet d'éviter d'effacer par erreur tel ou tel fichier par la commande erase par exemple. Il faut remettre cet attribut à -r pour l'inhiber.

L'attribut "caché" (ATTRIB toto.doc +h pour activé cet attribut sur le fichier toto.doc) permet au fichier de ne pas être affiché à l'occasion d'un listage de fichiers (commande DIR).

L'attribut "archive" indique si le fichier a été modifié depuis sa dernière sauvegarde. Cet attribut sert pour la commande XCOPY vue plus loin.

L'attribut "système" indique au système d'exploitation si tel ou tel fichier est un fichier faisant partie de ceux utilisés par le système d'exploitation. Dans la pratique nous n'avons pas à nous servir directement de cet attribut ni des fichiers réservés par MS-DOS.

---

## Gestion de dossiers

Voir le chapitre sur les [répertoires](#).

CD\ (Tapez CD + Alt 92). Retour au répertoire principal.

CD.. Retour au répertoire précédent.

CD WORD Ouvre le dossier WORD.

MD NOM Création du dossier NOM.

RD NOM Effacement d'un dossier. Au préalable on doit ouvrir le dossier et effacer tous les fichiers se trouvant dans ce dossier (y compris les sous répertoires), puis revenir à la racine comportant ce dossier.

DELTREE (MSDOS 6 et supérieur). Efface un répertoire même s'il contient des fichiers.

---

## Gestion des fichiers

Copie de fichiers (transfert d'un ou plusieurs fichiers, d'un lecteur à un autre ou d'un dossier à un autre) :

COPY C:NOM.EXT A: Copie d'un fichier se trouvant sur le lecteur C, vers le lecteur A et dont on précise le NOM et l'EXTension.

COPY C:\*.COM A: Copie de tous les fichiers se trouvant sur le lecteur C et dont l'extension est COM, sur le lecteur A.

COPY A:X\*.\* C: Copie de tous les fichiers se trouvant sur le lecteur A et dont les noms commencent par X, sur le lecteur C.

COPY A:.\*.\* C: Copie de tous les fichiers se trouvant sur le lecteur A, sur le lecteur C.

COPY A.TXT+B.TXT C.TXT Fusionne les fichiers A.TXT et B.TXT en un seul dans un troisième fichier C.TXT.

XCOPY A:\*. \* C:. Équivaut à la commande précédente (COPY) mais à la différence près qu'elle permet quelques options supplémentaires comme nous allons le voir.

XCOPY A:\*. \* C: /s. Copie de tous les fichiers se trouvant sur le lecteur A, sur le lecteur C mais en y incluant les fichiers situés dans les sous répertoires (option /s).

XCOPY A:\*. \* C: /s /m. Idem que la commande précédente mais cette fois ci en désactivant l'attribut archive (voir ci-dessus). Dans ce cas XCOPY (avec l'option /m) ne copie que les fichiers dont l'attribut archive est activé. Pour une dizaine de fichiers (dont la capacité totale serait de même pas 20 Ko) cela n'a certes pas grand intérêt. Mais là où l'on en sent l'utilité, c'est dans le cas d'un serveur hébergeant des milliers (voir des dizaines de milliers) de fichiers dont la capacité totale se chiffrent en centaines de millions d'octets. Il est évident que cela est beaucoup rapide de ne sauvegarder que les quelques dizaines de fichiers modifiés pendant la journée plutôt que d'être obligé de tout sauvegarder à chaque fois.

Changement de nom d'un fichier :

REN NOM.EXT NOUVEAUNOM.EXT

Effacement de fichiers :

ERASE NOM.EXT Effacement d'un fichier dont on précise le NOM et l'EXTension.

ERASE \*.BAK Effacement de tous les fichiers dont l'extension est BAK.

ERASE \*.\* Effacement de tous les fichiers. Dans ce cas, MS DOS nous demande de confirmer l'effacement en tapant O (oui) pour confirmer la demande d'effacement, ou N (non) pour annuler la demande.

---

### **Catalogue (listage des noms de fichiers stockés sur mémoire de masse)**

DIR Catalogue du lecteur en cours.

DIR A: Catalogue du lecteur A (idem pour B et C).

DIR \*.COM Catalogue de tous les fichiers dont l'extension est COM.

DIR X\*.\* Catalogue de tous les fichiers dont le nom commence par X.

Dans ce cas on obtient la liste de tous les fichiers avec leur coordonnées, taille, date et heure de la dernière sauvegarde des fichiers. Il existe deux options qui permettent de présenter différemment la liste de ces fichiers (/P et /W).

*Options de DIR*

DIR /S Affiche aussi le contenu des sous répertoire.

DIR /W Affiche uniquement le nom des fichiers sur cinq colonnes.

DIR /P Si le nombre de fichiers ne tient à l'écran, /P marque une pause avant d'afficher la suite.

DIR /B Donne uniquement le nom des fichiers sans leur coordonnées (taille et date).

On peut bien sûr combiner ces options. Exemple :

DIR \*.DOC /S /P Affiche les fichiers dont l'extension est DOC en y incluant les sous répertoires et en marquant une pause dans le défilement de la liste des fichiers.

---

## Visualisation du contenu d'un fichier texte

TYPE NOM.EXT Envoi à l'écran le contenu du fichier nom.ext. Ce fichier étant au format texte tel que les [batch](#) ou les .txt.

MORE < NOM.TXT Idem que précédemment mais à la différence que la visualisation marque une pause avant d'afficher la suite du texte si le fichier ne tient pas sur un écran.

---

## Impression

Après les commandes DIR et TYPE, si l'on précise la commande >PRN, cela enverra à l'imprimante la liste des fichiers (DIR) ou le contenu d'un fichier (TYPE) :

DIR >PRN. Impression de toute la liste des fichiers.

DIR \*.DOC>PRN. Impression de toute la liste des fichiers dont l'extension est DOC.

TYPE NOM.EXT>PRN. Impression du contenu d'un fichier dont on précise le NOM et l'EXTension.

---

## Lecteurs virtuels

On peut attribuer une lettre à un répertoire pour que celui-ci soit vu en tant que lecteur. Exemple, au lieu de faire dir c:\toto, on va pouvoir faire dir e:. Ici le répertoire c:\toto sera vu comme le lecteur e:. La commande pouvant faire cela est subst. Syntaxe générale : subst lecteur chemin.

### *Options*

**subst e: toto**, attribue la lettre e au dossier toto.  
**subst**, donne la liste des lettres associés aux chemins.  
**subst e: /d**, supprime l'association du lecteur e au dossier toto.

### *Le piège*

Quand on attribue une lettre à un répertoire, il faut prendre l'habitude de mettre le chemin complet du répertoire. Exemple : subst e: c:\toto.  
Un simple subst e: toto fonctionne certes pour attribuer la lettre e au répertoire toto. Mais avec l'inconvénient d'attribuer e au toto se trouvant dans le répertoire courant. Exemple, si l'on se trouve dans le répertoire document, subst e: toto attribue la lettre e au répertoire c:\document\toto et non à c:\toto.  
Quand on fera par exemple un copy a:\*. \* e:, les fichiers de la disquette se trouveront donc dans c:\document\toto et non dans c:\toto.

Sinon, inutile de dire que l'on peut mettre les subst nécessaires dans l'autoexec.bat si l'on souhaite une attribution automatique des lettres aux répertoires au démarrage de l'ordinateur.

---

## **Divers**

CLS. Effacement de l'écran.

---

## ***Les batch***

Les batch sont des fichiers qui permettent le traitement en lot des commandes msdos. S'il y a une suite de commandes que l'on doit régulièrement effectuer, on place ces commandes dans un fichier appelé batch. On peut donc y mettre toutes les commandes du type cls, cd, etc ; mais certaines instructions (que nous allons voir) sont spécifiques aux batch.

Ces fichiers batch ont l'extension ".bat" et sont au format texte, donc pouvant être créés avec n'importe quel éditeur de texte reconnaissant l'[ascii](#).

---

## Instruction echo

Cette instruction sert à deux choses, à afficher du texte à l'écran et à demander d'afficher ou non la suite des instructions à exécuter.

Le signe arrobas (@) permet de ne pas afficher l'instruction en cours d'exécution. Exemple : l'instruction "@copy toto titi" copie le fichier toto dans un fichier titi mais sans afficher l'instruction " copy toto titi ".

Pour éviter d'avoir à mettre l'arrobas à toutes les instructions, on peut mettre l'instruction @echo off. Cela demande aux batch de ne pas afficher les instructions qui suivent mais tout en les exécutant. L'instruction "@echo on" rétabli l'affichage.

L'instruction "@echo." affiche une ligne vide.

L'instruction "@echo toto" affiche le texte toto.

Exemple :

```
@echo off
cls
echo bonjour
echo.
echo toto
```

Résultat :

```
bonjour
toto
```

---

## Instruction exist

Exemple 1 :

```
if exist toto echo Le fichier toto existe.
if not exist toto echo Le fichier toto n'existe pas.
```

La première ligne (if exist) test si le fichier toto existe.

La deuxième ligne (if not exist) test si le fichier toto n'existe pas.

Dans les deux cas, le test if permet aussi d'afficher le message correspondant (echo...)

Exemple 2 :

```
if not exist rep md rep.
```

Si le répertoire "rep" n'existe pas, alors on le crée.

Exemple 3 :

```
if exist toto goto suite
rem Placer ici les instructions à effectuer si le fichier (ou le répertoire) toto n'existe
pas.
:suite
```

Permet ici de passer à la suite si le fichier toto existe, sinon d'effectuer des traitements en conséquence.

Ces exemples testent l'existence d'un fichier. Pour tester si un dossier existe, il faut mettre le paramètre \nul après le dossier à tester. Exemple pour tester si le dossier toto existe :

```
if exist toto\nul goto suite
echo n'existe pas
goto fin
:suite
echo existe
:fin
```

---

## Commande choice

La commande choice demande à l'utilisateur de saisir une des lettres proposées. La récupération du choix se fait par la commande errorlevel.

Options :

**/C[:]touches** : Spécifie les touches à proposer. Par défaut c'est ON (pour Oui ou Non).  
**/N** : Ne pas afficher les touches et le ? ... l'invite.  
**/S** : Permet de distinguer entre majuscules et minuscules.  
**/T[:]c,nn** : La touche "c" sera sélectionnée par défaut après nn secondes si aucune touche n'est appuyée.

Exemple (menu.bat) :

```
@echo off
:menu
cls
echo d) lancer defrag
echo s) lancer scandisk
echo q) quitter
```

```
choice /c:sdq Quelle application désirez-vous
if errorlevel = 3 goto fin
if errorlevel = 2 goto def
if errorlevel = 1 goto scan
:def
defrag
goto menu
:scan
scandisk
goto menu
:fin
```

Ce batch boucle jusqu'à ce que l'on veuille quitter.

---

## Les paramètres

La directive %1 (signe "pour cent" plus un numéro de 1 à 9) permet de traiter le paramètre transmis au fichier batch. On peut transmettre jusqu'à neuf paramètres aux fichiers batch.

Exemple : Le fichier batch x.bat contient echo %1. Si l'on tape x toto, le batch affichera toto.

Exemple de test :

```
if "%1"=="toto" goto xxx
goto fin
:xxx
echo toto a été transmis comme paramètre.
:fin
```

Attention, le test if distingue les majuscules des minuscules. Dans notre exemple, si l'on tape x Toto, le message en xxx ne sera pas exécuté ; il faut que toto soit entièrement saisi en minuscule.

---

## Commande shift

On veut copier des fichiers dont on précise les extensions (les \*.txt et les \*.doc par exemple). Mais pour corser le tout, on ne sait pas d'avance le nombre d'extensions que l'on veut traiter ni même les extensions. On suppose ici que l'on s'est placé dans le répertoire désiré et que l'on veuille copier les fichiers sur une disquette (lecteur A:).

Donnons d'abord le batch (ex.bat) que nous commenterons après :

```
:encore  
if "%1"==" " goto fin  
copy *.%1 a:  
shift  
goto encore  
:fin
```

Exemple de lancement : ex txt doc.

Commentaires :

Le test if commence par tester s'il n'y a rien en paramètre. Dans ce cas on termine le batch en allant à la fin.

Sinon :

On copie les fichiers dont l'extension se trouve dans le premier paramètre.  
La commande shift décale les paramètres d'un cran. C'est-à-dire que le contenu du deuxième paramètre se retrouve en premier, puis le troisième se place en deuxième, et ainsi de suite jusqu'au dernier. Le dernier paramètre se retrouve vide.  
Et enfin un goto encore permet de retourner au début du traitement.

Avec le test if on effectue les traitements jusqu'à ce que le premier paramètre soit vide.

---