

---

# Ligne de commande sous Windows

*BTS SIO - Bloc 1 - Support et mise à disposition des services informatiques*

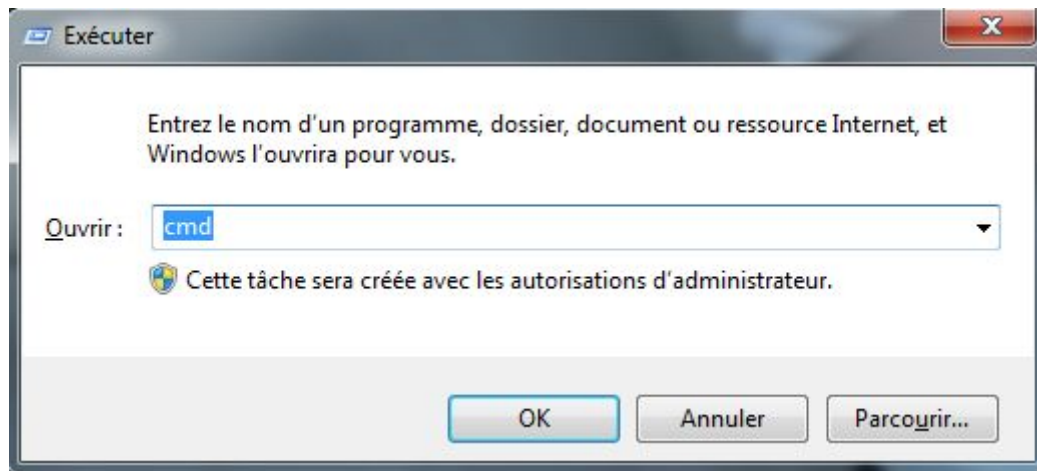
*U4 - 1.2 Répondre aux incidents et aux demandes d'assistance et d'évolution*

<b>1. Introduction - La ligne de commande sous Windows</b>	<b>3</b>
<b>2. Naviguer dans le système de fichiers</b>	<b>4</b>
2.1. Organisation des fichiers sur le disque	4
2.1.1. Notion de répertoire	4
2.1.2. Notion de lecteur courant et de répertoire courant	4
2.1.3. Chemin absolu, chemin relatif	4
2.1.4. Caractère générique	5
2.1.5. Répertoire personnel	5
2.2. Les principales commandes pour gérer les fichiers	5
<b>3. Modifier le système de fichiers</b>	<b>7</b>
3.1. Commandes à connaître	7
3.2. Redirections	7
<b>4. Les fichiers de commandes</b>	<b>8</b>
4.1. Principes	8
4.2. Paramétrage	8
4.3. Exécution depuis un langage de programmation	9

# 1. Introduction - La ligne de commande sous Windows

Windows dispose d'un mode ligne de commande hérité de son ancêtre MS-DOS.

Pour ouvrir un terminal sous Windows, cliquez sur le menu Démarrer et choisissez "Exécuter" (raccourci : Touche Windows + R). Tapez ensuite le texte cmd puis validez sur la touche Entrée.



## 2. Naviguer dans le système de fichiers

### 2.1. Organisation des fichiers sur le disque

#### 2.1.1. Notion de répertoire

Un disque dur peut contenir une très grande quantité de fichiers (en général plusieurs dizaines de milliers). Pour faciliter l'organisation, on va classer ces fichiers dans des dossiers (ou répertoires, directories en anglais).

Pour donner l'emplacement d'un fichier, on utilise cette représentation (c'est très important, car on utilisera cette représentation très souvent sous DOS, mais aussi dans n'importe quel langage de programmation) :

C:\Corel\Draw70\draw\Corel.exe

C:	Représente la lettre de lecteur sur lequel se trouve le fichier. Les lettres de lecteur sont généralement A: et B: pour les lecteurs de disquette, C: pour le disque dur, D: pour le lecteur de CD-ROM.
\	Signifie que le chemin que l'on indique ensuite commence à la racine du lecteur, et non pas à partir du répertoire courant (répertoire courant : expliqué plus loin).
Corel\Draw70\	Chemin dans lequel se trouve le fichier indiqué.
Corel.exe	Nom du fichier, composé du nom proprement dit et de l'extension EXE, qui signifie par convention que ce fichier est un programme exécutable.

Sous Windows, les différentes partitions de la machine sont identifiées par une lettre de lecteur. Historiquement, le premier disque dur correspond à la lettre C.

#### 2.1.2. Notion de lecteur courant et de répertoire courant



L'image ci-dessus est une invite de commande. Elle nous indique que :

- Le lecteur courant est le lecteur C:
- Le répertoire courant sur ce lecteur est \Users\r.varnier

#### 2.1.3. Chemin absolu, chemin relatif

L'emplacement de chaque ressource (fichier ou répertoire) dans le système de fichiers est appelé son chemin. Dans un chemin Windows, le séparateur dans entre deux répertoires est le caractère \ (antislash).

On distingue deux types de chemins :

- Un chemin absolu identifie une ressource en commençant à la racine de la partition, par exemple C:\. Un chemin absolu ne dépend pas du répertoire courant et est donc valide partout.
  - C:\Users\roger\hello.txt est un exemple de chemin absolu.
- Un chemin relatif identifie une ressource à partir du répertoire courant. Il dépend donc du répertoire courant et n'est pas valide partout.
  - ..\roger\adresses.txt (sans \ au début !) est un exemple de chemin relatif.

#### 2.1.4. Caractère générique

Le caractère générique \* (wildcard) permet de remplacer une partie d'un nom de fichier ou de répertoire. On l'utilise pour appliquer une commande à plusieurs éléments.

- copy f\*.txt monrep\ copie dans le répertoire monrep tous les répertoires dont le nom commence par un f et finit par .txt.
- del \* supprime tous les fichiers du répertoire courant.

#### 2.1.5. Répertoire personnel

Sous Windows, chaque utilisateur dispose d'un répertoire personnel à son nom situé dans C:\Users. Par exemple, le répertoire personnel de l'utilisateur nicolas est C:\Users\nicolas.

## 2.2. Les principales commandes pour gérer les fichiers

Commande	Rôle	Exemples d'utilisation
lecteur:	Permet de changer le lecteur courant.	C:  D:
DIR	Affiche le contenu d'un répertoire	DIR (affiche tous les fichiers du répertoire courant)  DIR A: (affiche tous les fichiers du répertoire courant de la disquette)  DIR A:*.exe (affiche tous les fichiers ayant l'extension EXE).
CD	Change le répertoire courant (Change Directory)	CD Command (descend dans le sous-répertoire Command).  CD .. (remonte d'un répertoire)
MD	Crée un répertoire (Make Directory)	MD Rep (crée le sous-répertoire nommé Rep)  MD C:\Rep (crée un sous-répertoire Rep à la racine du lecteur C:)

RD	Supprime un répertoire vide (Remove Directory)	RD Rep RD c:\Rep
COPY	Copie un ou plusieurs fichiers	COPY fichier_source fichier_destination
MOVE	Déplace un ou plusieurs fichiers	COPY fichier_source Chemin_destination
DEL	Supprime un ou plusieurs fichiers	DEL D:\*.*
REN	Renomme un fichier	REN ancien_nom nouveau_nom
TYPE	Affiche le contenu d'un fichier ascii	TYPE C:\Autoexec.BAT
EDIT	Affiche de contenu d'un fichier ascii dans l'éditeur, pour le modifier	EDIT C:\Autoexec.BAT
ATTRIB	Visualise les attributs d'un fichier, permet également de les modifier.	ATTRIB C:\Config.SYS
FORMAT	Permet de formater un disque	FORMAT C:

## 3. Modifier le système de fichiers

### 3.1. Commandes à connaître

- mkdir (make directory) crée un nouveau répertoire.
- mkdir monrep crée le répertoire monrep dans le répertoire courant.
- copy (équivalent de cp sous Linux) copie des fichiers ou des répertoires.
- copy fic1.txt monrep\ copie le fichier fic1.txt dans le répertoire monrep.
- copy fic1.txt fic2.txt duplique le fichier fic1.txt sous le nom fic2.txt.
- del (équivalent de rm sous Linux) supprime des fichiers.
- del fic1.txt supprime le fichier fic1.txt.

### 3.2. Redirections

Le caractère > permet de rediriger la sortie d'une commande vers un fichier en écrasant son contenu actuel.

Le caractère >> redirige la sortie d'une commande vers un fichier en l'ajoutant à la fin de son contenu actuel.

- echo Bonjour Monde > bonjour.txt remplace le contenu du fichier bonjour.txt par le texte "Bonjour Monde".
- echo Bonjour Monde >> bonjour.txt ajoute le texte "Bonjour Monde" à la fin du fichier bonjour.txt.

## 4. Les fichiers de commandes

### 4.1. Principes

Les fichiers de commandes permettent de créer des "macro-commandes", c'est-à-dire qu'on peut créer une nouvelle commande qui va être en fait une succession de commandes DOS élémentaires. On appelle ça le traitement "par lot" (Batch en anglais).

En fait, la suite de commandes à exécuter sera enregistrée dans un fichier ayant l'extension BAT. Pour exécuter cette suite de commandes, il suffira de taper le nom du fichier qui les contient.

Par exemple, si on dispose du fichier ACCUEIL.BAT suivant :

```
ECHO Bonjour  
DATE  
TIME
```

Alors le simple fait de taper la commande ACCUEIL.BAT (ou ACCUEIL) va exécuter toutes les commandes contenues dans ce fichier. ACCUEIL est donc une macro-commande.

**Terminologie** : les termes suivants sont synonymes : macro-commandes, fichier batch, fichier de commandes, fichier BAT (fait référence à l'extension de ces fichiers).

### 4.2. Paramétrage

De la même façon que les commandes MS-DOS reçoivent des paramètres, nos fichiers de commandes peuvent en recevoir également. Dans le fichier de commande, le symbole %1 va être remplacé par le premier paramètre, le symbole %2 va être remplacé par le deuxième paramètre, et ainsi de suite.

Par exemple, voici un script Archive.bat qui archive un fichier dont le nom est passé en paramètre, et désactive son attribut d'archive.

```
COPY %1 C:\Sauvegarde\  
ATTRIB -a %1  
ECHO Le fichier %1 a été archivé.
```

Voici le même script avec une petite variante : on passe en paramètre le nom du fichier à archiver, ainsi que le répertoire contenant les archives (on utilise donc deux paramètres au lieu d'un).

```
COPY %1 %2\  
ATTRIB -a %1  
ECHO Le fichier %1 a été archivé dans %2
```

Exemple d'appel du premier fichier de commandes où Archive est le nom de la macro-commande, MonFichier.TXT est le paramètre.

```
Archive MonFichier.TXT
```



Exemple d'appel du deuxième fichier de commandes où Archive2 est le nom de la macro-commande, MonFichier.TXT est le premier paramètre, D:\Sauvegarde est le deuxième paramètre.

```
Archive2 MonFichier.TXT D:\Sauvegarde
```

### 4.3. Exécution depuis un langage de programmation

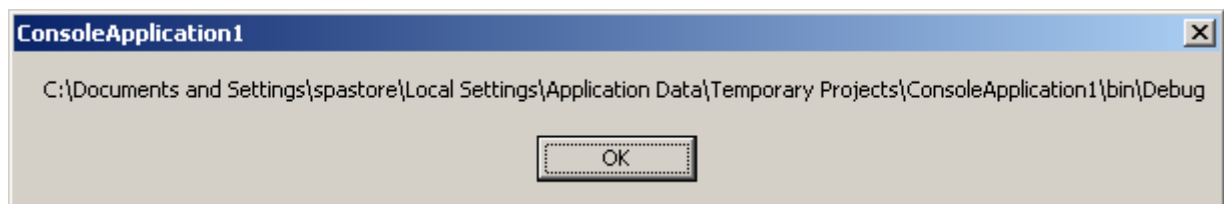
Exécuter des fichiers de commandes ou d'autres fichiers exécutables, à partir d'un langage de programmation, va nous permettre d'automatiser certaines tâches.

L'exécution d'un fichier à partir de Visual Basic.NET se fait grâce à l'instruction Shell :

```
Shell ("fichier de commandes ou exécutable, et son paramétrage")
```

Comme toujours, le fichier de commandes ou le fichier exécutable sera précédé du chemin d'accès vers ce fichier. Pour savoir quel est le répertoire courant lorsque vous exécutez une application Visual Basic, exécutez l'instruction Visual Basic suivante :

```
MsgBox(CurDir)
```



L'exécution d'un fichier à partir de PHP se fait grâce à l'instruction [exec](#) (ou [autres](#)) :

```
exec ("Fichier de commandes ou exécutable, et son paramétrage")
```