


La ligne de commande Windows et les fichiers batch

Par Victor Laurie  - Thomas Garcia 

Date de publication : 6 avril 2009

Dernière mise à jour : 10 novembre 2010

Cours complet sur la ligne de commande Windows et les fichiers de commandes (fichiers batch). Ce cours est une traduction du site  **commandwindows.com** dont l'auteur original est Victor Laurie. ♪

I - À propos de cet article.....	6
I-A - La traduction.....	6
I-B - Victor Laurie - L'auteur original.....	6
I-C - Remerciements.....	6
I-D - Remarques / Questions.....	6
II - Plan du cours.....	7
II-A - Introduction.....	7
II-B - Introduction à la ligne de commande.....	7
II-C - Applications spécifiques de la console de commande.....	7
II-C-1 - Gestion des fichiers.....	7
II-C-2 - Outils réseau et Internet.....	7
II-C-3 - Administration système et maintenance.....	8
II-D - Ajouts et extensions aux commandes natives.....	8
II-E - Fichiers Batch.....	8
II-F - Autres sujets concernant la ligne de commande.....	8
II-G - Vista.....	8
III - Introduction à l'invite de commande.....	9
III-A - DOS est mort, longue vie à la ligne de commande.....	9
III-B - La fenêtre d'invite de commande.....	9
III-C - Commandes internes et externes.....	9
III-D - Quelques symboles.....	10
IV - Liste des commandes.....	12
V - Les commandes utilisables par tous.....	16
V-A - Suppression avancée avec la commande « del ».....	16
V-B - Gestion avancée des dossiers (répertoires).....	17
V-C - Xcopy : Une commande puissante et polyvalente pour copier et sauvegarder les fichiers.....	17
V-D - Comment lister des fichiers avec « dir ».....	17
V-E - Renommer des fichiers avec « ren ».....	18
V-F - Déplacer des fichiers avec « move ».....	18
V-G - Changer les attributs de fichier avec « attrib ».....	18
V-H - Comment rendre vos dossiers favoris facilement accessibles dans les boîtes de dialogue Windows.....	19
V-I - Autres commandes.....	19
VI - Configurer la fenêtre d'invite de commande.....	20
VI-A - Introduction.....	20
VI-B - La boîte de dialogue des propriétés de l'invite de commande.....	20
VI-B-1 - Onglet Options.....	21
VI-B-2 - Tampon d'historique des commandes.....	21
VI-B-3 - Onglet Police.....	21
VI-B-4 - Onglet Configuration.....	21
VI-B-5 - Onglet Couleurs.....	22
VI-C - Application des changements à toutes les fenêtres d'invite.....	22
VI-D - Configurer des raccourcis pour l'invite de commande.....	23
VI-E - La commande « color ».....	23
VI-F - Options pour l'interpréteur de commande cmd.exe.....	24
VII - L'invite Démarrer-Exécuter.....	26
VII-A - Introduction à la fenêtre « Exécuter ».....	26
VII-B - Ouvrir des applications dans « Exécuter ».....	26
VII-B-1 - Ouvrir les consoles de gestion.....	27
VII-B-2 - Ouvrir les éléments du panneau de configuration.....	28
VII-B-3 - Rundll32.exe.....	28
VII-C - Ouvrir des dossiers dans « Exécuter ».....	28
VII-D - Glisser-Déplacer des dossiers et des fichiers dans la fenêtre « exécuter ».....	28
VII-E - Ajouter des applications au « path ».....	28
VII-F - Applications Internet.....	29
VII-G - Comparaison entre la fenêtre « Exécuter » et l'invite de commande.....	29
VII-H - Accéder à la console de commande depuis la fenêtre « Exécuter ».....	29
VII-I - La fenêtre « Exécuter » dans Vista.....	30
VIII - Gestion des fichiers à partir de la ligne de commande - Assoc et Ftype.....	31

VIII-A - Introduction.....	31
VIII-B - Gestion des types de fichiers et des extensions avec la commande « assoc ».....	31
VIII-C - Gérer les types de fichiers et les associations de programmes avec la commande « ftype ».....	32
IX - Xcopy et ses applications.....	33
IX-A - Syntaxe de la commande Xcopy.....	33
IX-B - Applications de Xcopy.....	35
IX-C - Exclure des fichiers et dossiers.....	35
IX-D - Codes retour de Xcopy.....	36
X - Les commandes réseau « net » dans Windows XP.....	37
X-A - Les commandes « net ».....	37
X-B - Commande Net User dans Windows Vista.....	37
XI - Netstat.....	38
XI-A - Syntaxe et options.....	38
XI-B - Applications de Netstat.....	39
XI-B-1 - Vérifier les connexions TCP/IP.....	39
XI-B-2 - Rechercher les programmes malveillants en regardant quels programmes ouvrent des connexions.....	40
XI-B-3 - Programme batch pour vérifier les connexions et s'arrêter automatiquement.....	41
XII - Netsh, la console des services Réseau.....	42
XII-A - Introduction à Netsh.....	42
XII-B - Le contexte « netsh diag ».....	42
XII-B-1 - L'interface graphique.....	43
XII-B-2 - Le contexte « netsh interface ip ».....	43
XII-B-3 - Réinitialiser le protocole Internet (TCP/IP).....	44
XII-C - Le contexte « netsh winsock ».....	44
XII-D - Netsh Firewall Helper dans Windows XP SP2 (assistance pare-feu).....	44
XIII - TCP/IP et outils Réseau.....	46
XIII-A - Outil de configuration IP de Windows (ipconfig).....	46
XIII-B - Options d'IPConfig.....	47
XIII-C - Ping.....	48
XIII-D - Tracert.....	48
XIII-E - Pathping.....	49
XIII-F - Netstat.....	49
XIII-G - Nslookup.....	49
XIII-H - Netsh.....	50
XIV - Utilitaire de système de fichiers - Fsutil.exe.....	51
XIV-A - Sous-commandes.....	51
XIV-B - Exemples d'applications de Fsutil.....	52
XIV-B-1 - Utilisations de la sous-commande « behavior ».....	52
XIV-B-1-a - Consultez les réglages actuels.....	53
XIV-B-1-b - Désactiver les noms de fichiers courts pour accélérer Windows.....	53
XIV-B-1-c - Désactivation de l'heure de dernière modification pour accélérer Windows.....	53
XIV-B-1-d - Changer la taille de la MFT (Master File Table).....	54
XIV-B-2 - Afficher quelques propriétés du lecteur.....	54
XIV-B-2-a - Obtenir la liste des lecteurs d'un ordinateur.....	54
XIV-B-2-b - Obtenir le type de lecteur.....	54
XIV-B-2-c - Obtenir des informations générales sur un volume.....	54
XIV-B-2-d - Obtenir la quantité d'espace libre sur un lecteur.....	54
XV - La console de récupération de Windows XP.....	56
XV-A - Accéder à la console de récupération avec un CD de Windows.....	56
XV-B - Accéder à la console de récupération sans un CD de Windows.....	56
XV-C - Commandes dans la console de récupération.....	57
XV-D - Enlever les restrictions de la console.....	58
XV-D-1 - Utilisation de la console de stratégie de sécurité locale ou de stratégie de groupe.....	59
XV-D-2 - Modifier le Registre.....	59
XV-D-3 - Un script pour effectuer la modification du Registre.....	60
XV-E - Utilisation de la commande « Set ».....	60
XV-F - La seule commande à apprendre.....	61

XVI - Commandes de la console de récupération.....	62
XVII - Gérer le Registre Windows à partir de l'invite de commande avec « Reg.exe ».....	67
XVII-A - Reg add.....	68
XVII-B - Reg delete.....	69
XVII-C - Sauvegarder et restaurer le Registre.....	69
XVII-C-1 - Sauvegarder et restaurer des fichiers ruches (hives).....	69
XVII-C-2 - Exporter et importer des fichiers REG.....	70
XVII-D - Reg Query.....	70
XVIII - Gérer les Services Windows avec la commande de contrôle des services SC.....	71
XVIII-A - Les sous-commandes de SC.....	71
XVIII-B - Exemples d'applications utiles de SC.....	71
XIX - Gérer les programmes de Windows à partir de l'invite de commande : Tasklist.....	74
XIX-A - Commande Tasklist basique.....	74
XIX-B - Plus d'options avancées pour Tasklist.....	75
XIX-B-1 - Trouver quels services utilisent un processus.....	76
XIX-B-2 - Trouver quels fichiers DLL sont utilisés par un processus.....	76
XIX-C - Filtrer le résultat de Tasklist.....	77
XX - Gérer les programmes de Windows : Tskill et Taskkill.....	79
XX-A - Tskill.....	79
XX-B - Taskkill.....	80
XX-C - Filtrer le résultat de Taskkill.....	81
XX-D - Exemple d'utilisation des filtres dans Taskkill.....	82
XXI - Exécuter des fichiers VBScripts et JScript à partir de l'invite de commande.....	83
XXI-A - Configurer l'invite de commande pour exécuter des scripts.....	83
XXI-B - Quelques scripts fournis avec Windows XP.....	83
XXI-C - Outil de ligne de commande WMI : WMIC (Windows Management Instrumentation Command-line)....	84
XXII - Outils utiles pour Windows XP provenant du Resource Kit de Windows Server 2003.....	85
XXIII - Support Tools de Windows XP SP2.....	89
XXIII-A - Une sélection des Support Tools de Windows XP en ligne de commande.....	89
XXIV - Les fichiers Batch (scripts) dans Windows.....	91
XXIV-A - Que-est-ce qu'un fichier batch ?.....	91
XXIV-B - Construire un fichier Batch.....	91
XXIV-B-1 - Une version plus générale avec des arguments.....	92
XXIV-B-2 - L'instruction Rem.....	93
XXIV-B-3 - Plus d'exemples.....	93
XXIV-B-4 - Demander une saisie à l'utilisateur.....	93
XXIV-C - Pour en savoir plus.....	94
XXV - Des fichiers batch plus puissants : Partie 1 - Branchements et boucles.....	95
XXV-A - Branchement conditionnel avec l'instruction « if ».....	95
XXV-A-1 - Instruction « if exist ».....	95
XXV-A-2 - Instruction « if defined ».....	95
XXV-A-3 - Instruction « if errorlevel ».....	96
XXV-A-4 - Opérateurs de comparaison.....	96
XXV-B - La commande « goto ».....	97
XXV-B-1 - Le label "End of File (:eof) est utilisé pour sortir d'un script.....	97
XXV-C - Boucles avec « if » et « goto ».....	97
XXVI - Des fichiers batch plus puissants : Partie 2 - Itérations avec « For ».....	99
XXVI-A - Itération simple dans une liste.....	100
XXVI-B - Boucler sur une série de valeurs.....	100
XXVI-C - Utiliser les répertoires.....	100
XXVI-D - Parcourir récursivement les sous-répertoires.....	101
XXVI-E - Parcourir des fichiers, chaînes et sorties de commandes.....	101
XXVII - Variables dans l'invite de commandes Windows.....	102
XXVII-A - Comment les variables sont définies avec la commande « set ».....	102
XXVII-A-1 - Localiser les variables.....	102
XXVII-A-2 - Variables saisies par l'utilisateur.....	103
XXVII-B - Opérations arithmétiques.....	103
XXVII-C - Instruction de comparaison de variables dans les fichiers batch.....	103

XXVIII - Doskey dans l'invite de commandes Windows XP.....	105
XXVIII-A - Syntaxe et fonctions.....	105
XXVIII-B - Plus d'informations.....	105
XXIX - PowerShell dans Windows XP.....	106
XXIX-A - Introduction à PowerShell.....	106
XXIX-B - Les fonctionnalités de PowerShell.....	106
XXIX-B-1 - Liste de cmdlets.....	106
XXIX-B-2 - Syntaxe des cmdlets.....	107
XXIX-C - Scripting PowerShell.....	108
XXIX-D - Plus d'informations.....	108
XXX - Astuces pour l'utilisation de l'invite de commande Windows.....	109
XXX-A - Comment afficher une ligne vide dans un fichier batch.....	109
XXX-B - Forcer « echo » à ne pas parcourir les arguments.....	109
XXX-C - Vérifier si un fichier existe.....	109
XXX-D - L'utile périphérique « nul ».....	110
XXX-E - Arrêter une commande en cours d'exécution.....	110
XXX-F - Mettre en pause le défilement de l'écran.....	110
XXX-G - Utiliser le glisser-déplacer.....	110
XXX-H - Remonter au répertoire parent.....	110
XXX-I - Comment changer le répertoire de travail courant vers un répertoire d'un autre disque.....	111
XXX-J - Attention aux espaces dans les noms des fichiers et des dossiers.....	111
XXX-K - Traitement spécial des variables pour l'instruction « for » dans les fichiers batch.....	111
XXX-L - Variables sensibles à la casse dans les instructions « for ».....	111
XXX-M - Mettre un raccourci vers une ligne de commande dans le Menu Démarrer.....	111
XXX-N - Créer un raccourci vers une commande.....	111
XXX-O - Ouvrir l'explorateur Windows à partir de la ligne de commande.....	112
XXX-P - Utilisation de la commande « Start ».....	112
XXX-Q - Utilisez la complétion de nom de fichier et de nom de dossier.....	112
XXX-R - Activer le mode d'édition rapide pour la fenêtre de commande.....	112
XXX-S - Afficher l'historique des commandes.....	113
XXX-T - Utiliser la commande « sleep » dans les fichiers batch sous Windows XP.....	113
XXX-U - Copier du texte à partir de la fenêtre de commande.....	113
XXX-V - Astuces pour la console de commande de Vista.....	113
XXXI - Accéder aux répertoires systèmes avec la console de commande dans Windows Vista.....	114
XXXI-A - Dossiers systèmes de Windows Vista.....	114
XXXII - Liste des commandes de Windows Vista et Référence.....	116
XXXII-A - Commandes dans Windows Vista.....	116
XXXIII - Astuces pour utiliser la console de commandes de Vista.....	121
XXXIII-A - Exécuter en tant qu'administrateur dans Vista.....	121
XXXIII-B - « Ouvrir une invite de commande ici » dans Vista.....	121
XXXIII-C - Ouvrir une fenêtre de commandes avec les privilèges administrateurs à partir de n'importe où.....	121
XXXIII-D - Envoyer la sortie d'une commande vers le presse-papier Windows avec clip.exe.....	121
XXXIII-E - Placer le contenu d'un fichier texte dans le presse-papier Windows avec clip.exe.....	122
XXXIII-F - La commande « choice » est de retour dans Vista.....	122
XXXIII-G - Utiliser la commande « choice » pour faire attendre un fichier batch sous Vista.....	122
XXXIII-H - Utiliser la commande « timeout » pour faire attendre un fichier batch dans Vista.....	122
XXXIII-I - Commutateur ajouté à « Dir » pour activer l'affichage des Flux de Données Alternatifs (Alternate Data Streams).....	122
XXXIII-J - Activer le compte administrateur principal sur l'écran de connexion.....	122
XXXIII-K - Réduire l'espace utilisé par la Restauration Système.....	123
XXXIII-L - Augmenter le cache mémoire du système de fichier.....	123
XXXIII-M - Utiliser « takeown » pour accéder à certains fichiers.....	123
XXXIII-N - Obtenir les droits de contrôle sur un fichier avec « icacs ».....	124
XXXIII-O - Nettoyer les fichiers de Vista SP1.....	124
XXXIII-P - Utilisation du Drag&Drop (Glisser-Déplacer).....	124

I - À propos de cet article

I-A - La traduction

Ce cours est une traduction du site [commandwindows.com](http://www.commandwindows.com) réalisée par **Thomas Garcia(Shawn12)** pour Developpez.com avec l'aimable autorisation de son auteur Victor Laurie.

La traduction a été réalisée à l'aide du logiciel gratuit d'aide à la traduction **OmegaT**.
La mise en forme au gabarit de Developpez.com a été effectuée grâce à l'éditeur écrit par **nono40**.

I-B - Victor Laurie - L'auteur original



Victor Laurie

Victor Laurie est un professeur de chimie à la retraite qui est instructeur bénévole pour des cours d'informatique au SeniorNet Learning Center à Ewing, NJ (USA). Il aide également à maintenir les ordinateurs en fonctionnement. Au cours des dernières années, il a préparé des présentations PowerPoint et Macromedia Flash pour les utiliser dans ses cours ainsi que pour aider les autres membres du corps enseignant. Il réalise également des présentations et rédige des articles sur des sujets informatiques pour différentes organisations à Princeton, NJ (USA).

I-C - Remerciements

Un grand merci tout d'abord à l'auteur original de l'article, Victor Laurie, qui nous a gracieusement accordé les droits pour cette traduction.

Merci également à **Laurent Dardenne**, **ram-0000**, **Vow** et **Toulouseing** pour leur relecture minutieuse et leur correction orthographique.

I-D - Remarques / Questions

Vous pouvez donner votre avis sur cet article en laissant un commentaire dans [cette discussion sur le forum](#).
Si vous rencontrez un problème technique avec l'invite de commande Windows ou les fichiers batchs, vous pouvez visiter le forum [Script/Batch](#).

II - Plan du cours



Ce chapitre est une traduction de la page  <http://commandwindows.com/> dont l'auteur original est Victor Laurie.

II-A - Introduction

La ligne de commande Windows est un élément essentiel pour les administrateurs système, mais est relativement méconnue de nombreux utilisateurs de PC. Le but de cet article est de rendre plus accessibles à la grande majorité des utilisateurs la puissance et l'utilité de la ligne de commande. Les fichiers batch et les scripts, souvent sous-estimés, seront également exposés ici.

Windows possède une interface graphique bien connue, mais beaucoup d'utilisateurs ont tendance à oublier (ou ne savent pas) qu'il existe des fonctions très utiles en ligne de commande. En fait, dans Windows, il y a beaucoup plus que le simple « pointer, cliquer ». Le clavier et la ligne de commande peuvent être un apport intéressant à la souris et aux icônes. Il y a deux fonctionnalités qui utilisent les lignes de commande. La première est l'outil « Exécuter » (ou zone de recherche dans Windows Vista) du Menu Démarrer. La deuxième est l'invite de commande Windows.

II-B - Introduction à la ligne de commande

Ceux qui découvrent la ligne de commande ou ont besoin d'une remise à niveau peuvent lire les éléments de base dans ces chapitres :

- [Ligne de commande - Introduction](#)
- [Liste des commandes et référence](#)
- [Les commandes utilisables par tous](#)
- [Configurer la fenêtre d'invite de commande](#)
- [Invite Démarrer->exécuter](#)

II-C - Applications spécifiques de la console de commande

Les chapitres indiqués ci-dessous donnent des détails et des exemples sur quelques commandes. Les sujets traitent de la maintenance d'ordinateur, l'administration système, la gestion de fichiers, les outils Internet et l'administration réseau.

II-C-1 - Gestion des fichiers

- [Assoc](#)
- [Ftype](#)
- [Xcopy](#)

II-C-2 - Outils réseau et Internet

- [Net Services \(Net\)](#)
- [Netstat](#)
- [Console des services réseau \(Netsh\)](#)
- [Outils réseau TCP/IP](#)

II-C-3 - Administration système et maintenance

- **Utilitaire de système de fichiers - Fsutil**
- **Console de récupération**
- **Commandes de la console de récupération**
- **Console de modification du Registre**
- **Commande de contrôle des services (SC)**
- **Tasklist**
- **Taskkill**
- **Tskill**
- **Xcopy**

II-D - Ajouts et extensions aux commandes natives

- **Scripts en ligne de commande**
- **Outils Windows serveur 2003 pour XP**
- **Outils de support (Support tools)**

II-E - Fichiers Batch

Les fichiers batchs fournissent un moyen simple d'effectuer des tâches répétitives ou longues à exécuter. Bien que les fichiers batch puissent être un peu sophistiqués, les bases sont assez simples pour être utiles à tous les utilisateurs qui ne connaissent pas la programmation.

- **Introduction aux fichiers Batch**
- **Branchements et boucles avec « if » et « goto »**
- **Itérations et boucles avec « For... in...do »**
- **Les variables et la commande « Set »**

II-F - Autres sujets concernant la ligne de commande

- **Doskey**
- **PowerShell**
- **Astuces pour utiliser l'invite de commande**

II-G - Vista

- **Shell command**
- **Liste des commandes Vista**
- **Astuces de la ligne de commande Vista**

III - Introduction à l'invite de commande



Ce chapitre est une traduction de la page  <http://commandwindows.com/command1.htm> dont l'auteur original est Victor Laurie.


III-A - DOS est mort, longue vie à la ligne de commande

Avant de continuer, je voudrais dissiper un malentendu selon lequel la ligne de commande serait la même chose que DOS. Même certains experts appellent l'invite de commande « fenêtre DOS ». En fait, Microsoft elle-même ne fait pas toujours la distinction. Il y a des ressemblances superficielles et certaines commandes ont le même nom, mais en réalité, le vieux DOS 16bits est mort. Tous les reliquats du DOS ont totalement disparu du noyau de Windows XP (il y a un émulateur DOS pour les anciens programmes). Windows XP est un système 32 bits à mémoire protégée avec une approche totalement différente de celle de la famille DOS/9X/Me. La ligne de commande dans XP possède beaucoup plus de possibilités et aucune limitation 16 bits par exemple le format DOS des noms de fichier (8.3). L'augmentation des possibilités fait de la ligne de commande un puissant outil.

III-B - La fenêtre d'invite de commande

L'invite de commande est exécutée depuis sa propre fenêtre en appelant l'interpréteur de commande de Windows XP qui est fourni par le fichier `cmd.exe` (situé dans le dossier **Windows\System32**). (L'ancien interpréteur de commande DOS est `command.com`.) Si vous regardez dans ce dossier vous pourrez également voir plusieurs fichiers qui semblent suspects ressemblant à certains vieux fichiers DOS. Ce sont toutefois des versions 32-bits avec de nombreuses nouvelles fonctionnalités. La fenêtre d'invite de commande peut être ouverte en tapant « `cmd` » (sans guillemets) dans **Démarrer-->Exécuter** ou via **Démarrer --> Tous les programmes -->Accessoires**. Une fenêtre en noir et blanc s'ouvre alors sur l'invite de commande (les couleurs peuvent être changées). La fenêtre ressemble beaucoup à l'ancien DOS, mais ne vous y trompez pas, ce n'est pas la même chose. Il est possible d'ouvrir plusieurs fenêtres d'invite de commande, celles-ci s'exécutant indépendamment. Il est même possible d'exécuter un interpréteur de commandes à l'intérieur d'un autre interpréteur de commande.

III-C - Commandes internes et externes

Il existe deux sortes de commandes pouvant être exécutées dans l'invite de commande. Il y a les commandes internes, directement intégrées dans l'interpréteur de commandes comme « `del` » et « `dir` ». Ces commandes peuvent être uniquement exécutées depuis une invite de commande (ou en appelant l'interpréteur de commande d'une autre façon). Elles sont listées dans le tableau ci-dessous. Il existe également une grande quantité de commandes externes qui utilisent un fichier exécutable additionnel. Cet exécutable peut être appelé depuis l'invite de commande ou bien via **Démarrer --> Exécuter**. Les détails concernant ces différentes commandes sont disponibles en plusieurs endroits. Dans Windows XP Professionnel, un fichier d'aide `ntcmds.chm` décrit les détails concernant les commandes ainsi que leurs options. Ce fichier d'aide peut être ouvert en tapant (sans les guillemets) « `hh ntcmds.chm` » dans **Démarrer --> Exécuter**. Cette aide peut être présente ou non dans Windows XP édition Familiale suivant l'installation effectuée. Cependant, vous pouvez obtenir une liste des commandes disponibles (non exhaustive) en tapant « `help` » (sans les guillemets) dans une invite de commande. Pour des détails sur l'une des commandes en particulier, il faut saisir « `help nom-de-commande` » ou « `nom-de-commande /?` ». Par exemple, si vous souhaitez obtenir de l'aide sur la commande `xcopy`, saisissez « `help xcopy` » ou « `xcopy /?` ». Les ressources Microsoft sont parfois déplacées, mais à la dernière vérification, une référence sur la ligne de commande est disponible  [sur ce lien](#) (en anglais).

assoc	dir	move	set
break	echo	path	setlocal
call	endlocal	pause	shift
cd	exit	popd	start
cls	for	prompt	time
color	ftype	pushd	title
copy	goto	rd	type
date	if	rem	verify
del	md	ren	volume

III-D - Quelques symboles

En plus des commandes, il y a plusieurs symboles qui peuvent être utilisés. Ils modifient ou combinent l'action des commandes. Ils sont listés dans le tableau ci-dessous.

Symbole	Fonction	Exemple
>	Envoie la sortie dans le fichier spécifié. Si le fichier existe, il est écrasé.	commande > fichier
>>	Ajoute la sortie à la fin du fichier spécifié. Le fichier est créé s'il n'existe pas.	commande >> fichier
<	Utilise le contenu du fichier spécifié en entrée de la commande.	commande < fichier
	Envoie la sortie de la commande1 dans l'entrée de la commande2	commande1 commande2
&	Utilisé pour combiner deux commandes. Exécute commande1 puis commande2.	commande1 & commande2
&&	Utilisé pour combiner deux commandes avec condition. Exécute commande2 uniquement si commande1 s'est terminée avec succès.	commande1 && commande2
	commande2 est exécutée uniquement si commande1 ne s'est pas exécutée correctement.	commande1 commande2
@	S'utilise en début de ligne dans un fichier batch pour ne pas afficher la commande.	@commande

Les symboles les plus couramment utilisés sont les deux symboles de redirection : « > » et « >> » ainsi que le symbole appelé « pipe » : « | » (Le symbole « pipe » est le symbole situé sur la touche 6 (au-dessus des lettres) des claviers AZERTY accessible par la combinaison [Alt Gr]+[6].) Les redirections sont fréquemment utilisées pour sauvegarder la sortie des commandes dans un fichier. Par exemple, la commande :

```
dir rep > fichier.txt
```

écrit la liste des fichiers du répertoire « rep » dans « fichier.txt ». Cette utilisation est décrite sur cette page. Le « pipe » est souvent utilisé pour contrôler l'affichage de certaines commandes produisant beaucoup de données en sortie. Par exemple si vous voulez vérifier la liste des fichiers d'un répertoire contenant un très grand nombre d'éléments, vous pouvez afficher une page après l'autre en établissant un « pipe » vers la commande more :

```
dir repertoire | more
```

IV - Liste des commandes




Ce chapitre est une traduction de la page <http://commandwindows.com/command3.htm> dont l'auteur original est Victor Laurie.



La liste complète des commandes possibles étant assez longue, les commandes listées ici sont une sélection de celles qui, je pense, sont les plus fréquemment utilisables pour un usage sur un ordinateur personnel.

Plusieurs des commandes listées ci-dessous sont décrites plus en détail plus loin dans cet article (indiqué par un lien « Voir les détails »). La plupart des commandes possèdent des options qui ne sont pas indiquées dans le tableau. Pour plus d'information, ouvrez une invite de commande et saisissez « nom-de-commande /? » (sans guillemets). Des informations détaillées sur ces commandes ainsi qu'une liste plus complète dans [la référence Microsoft](#) (en anglais).

Commande	Description	Exemple
assoc	Affiche ou modifie les associations de fichier. Utilisée seule, affiche une liste de toutes les associations de fichiers actuellement paramétrées.	Voir les détails
at	Planifie l'exécution de commandes ou de programmes sur un ordinateur à une heure et une date donnée. Nécessite le service de planification. Remplacé par <i>schtasks</i> .	
attrib	Configure les attributs de fichiers (<i>lecture seule, caché, système</i>)	Voir les détails
bootcfg	Utilisé pour réparer ou éditer le fichier de démarrage <i>boot.ini</i>	Voir les détails
cd ou chdir	Affiche le nom du répertoire courant ou change le répertoire courant	<code>cd chemin</code>
chkdsk	Recherche des erreurs sur les disques durs. Permet	Voir les détails

	de les réparer avec certains commutateurs.	
cls	Efface le contenu de la fenêtre d'invite de commande	cls
copy	Copie un fichier d'un emplacement à un autre	copy <i>fichier-sourcerepertoire-cible</i>
del	Supprime un ou plusieurs fichiers	Voir les détails
dir	Affiche la liste des fichiers et sous-dossiers contenus dans le répertoire.	Voir les détails
echo	Utilisé pour afficher un message ou pour activer/désactiver l'affichage des messages dans un script batch	echo message
exit	Quitte le programme cmd.exe(interpréteur de commandes) ou le script batch courant.	exit
fc	Affiche ou modifie les types de fichiers utilisés dans les associations d'extension de fichiers	fc fichier1 fichier2
for	Exécute une commande donnée pour chaque fichier d'un ensemble de fichiers.	Voir les détails
fsutil	Permet d'afficher et configurer certains paramètres du système de fichiers	Voir les détails
ftype	Affiche ou modifie les types de fichiers utilisés dans les associations d'extension de fichiers	Voir les détails
getmac	Affiche l'adresse MAC (Media Access Control) de votre carte réseau	getmac
goto	Dirige cmd.exe sur une ligne étiquetée dans le programme de commandes.	Voir les détails
if	Lance l'exécution conditionnelle dans un programme de commandes.	Voir les détails
ipconfig	Affiche toutes les valeurs de la configuration réseau TCP/IP courante et rafraichit les réglages DHCP (Dynamic Host Configuration Protocol et	Voir les détails

	DNS (Domain Name System)	
md ou mkdir	Crée un répertoire ou un sous-répertoire	Voir les détails
more	Affiche une page écran à la fois. Utilisé avec une autre commande ayant beaucoup de données en sortie	<i>commande more</i>
move	Déplace un fichier d'un dossier à un autre.	Voir les détails
net	Une suite de plusieurs commandes de configuration du réseau	Voir les détails
netsh	Console des services réseau. Suite de commandes complexes.	Voir les détails
netstat	Affiche les connexions TCP actives, les ports que l'ordinateur écoute, statistiques Ethernet, table de routage IP, statistiques pour les protocoles IP, ICMP, TCP, et UDP	Voir les détails
path	Définit le contenu de la variable d'environnement PATH qui contient la liste des répertoires utilisés pour chercher les fichiers exécutables	 Voir les détails
pathping	Fournit des informations sur les performances du réseau et les conditions au niveau des points intermédiaires entre la source et la destination	Voir les détails
pause	Commande utilisée pour suspendre l'exécution d'un script batch jusqu'à ce que l'utilisateur presse une touche	<i>pause</i>
ping	Vérifie la connectivité vers un ordinateur, un routeur ou un site Internet	Voir les détails
popd, pushd	Change le répertoire référencé dans l'invite de commande. Pushd change le répertoire et stocke le répertoire précédent. Popd change le répertoire courant par celui stocké par la commande pushd	<i>pushd repertoire</i> <i>popd</i>
powercfg	Gère les réglages d'alimentation comme	

	la veille prolongée (hibernation).	
reg	Ajoute, modifie et affiche les entrées du registre.	Voir les détails
rd ou rmdir	Supprime un répertoire	Voir les détails
ren ou rename	Change le nom d'un fichier ou d'un ensemble de fichiers	Voir les détails
sc	Commande utilisée pour obtenir des informations sur les services et pour les configurer	Voir les détails
schtasks	Planifie des commandes ou des programmes pour qu'ils s'exécutent périodiquement ou à une heure spécifique	
set	Affiche, définit ou supprime les variables d'environnement	Voir les détails ici et  ici
sfc	Scanne les fichiers systèmes et vérifie les versions de tous les fichiers systèmes protégés	sfc /scannow
shutdown	Arrête ou redémarre un ordinateur	 Voir les détails
start	Lance une application ou ouvre une nouvelle fenêtre d'invite de commande	Voir les détails
subst	Associe un répertoire à une lettre de lecteur	Voir les détails
systeminfo	Affiche des informations détaillées sur la configuration d'un ordinateur et sur son système d'exploitation	systeminfo more
taskkill	Termine un ou plusieurs tâches ou processus	Voir les détails
tasklist	Affiche une liste des applications et services avec leur PID (Process ID) pour toutes les tâches en cours d'exécution	Voir les détails
tree	Affiche graphiquement la structure d'un répertoire ou d'un disque	tree <i>repertoire</i>
type	Affiche le contenu d'un fichier texte	type <i>fichier.txt</i>
xcopy	Commande puissante avec de nombreuses options pour copier et sauvegarder des fichiers ou des répertoires	Voir les détails

V - Les commandes utilisables par tous



Ce chapitre est une traduction de la page <http://commandwindows.com/command2.htm> dont l'auteur original est Victor Laurie.

L'invite de commande est souvent utilisée par les administrateurs système, c'est pourquoi beaucoup de commandes sont spécialisées pour l'administration et les réseaux. Cependant, les utilisateurs lambda ne sont pas oubliés : il existe des commandes puissantes utilisables pour des tâches de tous les jours. Voici des exemples pour illustrer ces commandes.

Par commodité dans les exemples ci-dessous, je vais utiliser des noms simples pour les fichiers et les répertoires sans indiquer le chemin complet. Rappelez-vous que les chemins pour toutes les commandes sont relatifs au répertoire indiqué directement par l'invite de commande, dans le cas contraire, cela sera indiqué. Le répertoire de référence par défaut est généralement « C:\Documents and Settings\(\nom-de-connexion)\ ». Ainsi, si vous faites référence à un fichier « fichier.txt », l'ordinateur supposera que vous voulez utiliser « C:\Documents and Settings\(\nom-de-connexion)\fichier.txt ». Si le fichier que vous voulez est « C:\unrepertoire\fichier.txt », alors vous devez indiquer le nom complet avec le chemin correct. Attention, lorsque le chemin contient des espaces il doit être entouré par des guillemets.

Il existe un moyen pratique pour pouvoir ouvrir l'invite de commande directement dans le dossier de votre choix, il suffit d'installer le PowerToy fourni par Microsoft : « Command Prompt Here ». Un clic droit sur un dossier vous proposera d'accéder directement à l'invite de commande dans le dossier choisi. Cet outil peut être téléchargé à [cette adresse](#). Une autre méthode pour avoir un nom de fichier dans la console sans avoir à saisir un long chemin est d'utiliser le Glisser-Déplacer. Ouvrez une invite de commande et saisissez la commande que vous voulez utiliser suivie d'un espace. Utilisez ensuite l'explorateur Windows pour ouvrir le répertoire contenant le fichier que vous voulez utiliser. Faites glisser le fichier voulu dans la fenêtre d'invite de commande. (Cette fonctionnalité ne fonctionne pas dans Windows Vista)

V-A - Suppression avancée avec la commande « del »

La commande de suppression « del » possède maintenant un commutateur qui permet la suppression des fichiers dans les sous-répertoires. Ainsi, la commande

```
del /s monDossier\*
```

va supprimer tous les fichiers dans « monDossier » ainsi que tous les fichiers dans tous les sous-dossiers de « monDossier ». Remarquez le joker «*» qui permet des suppressions multiples en une seule opération. Utilisée avec l'option « /s », un simple « del » permet de vider le répertoire Temp ou pour faire le ménage sur le disque.

Une autre option « /f » permet de forcer la suppression des fichiers en lecture seule.

Si vous êtes sûr de ce que vous êtes en train de supprimer, ajouter le commutateur « /q » pour que la commande ne vous demande pas de confirmation pour chaque fichier. *En raison de la puissance de la commande « del », elle doit être utilisée avec beaucoup de précautions. Les fichiers supprimés par cette commande ne vont pas dans la Corbeille, mais sont supprimés définitivement.*

Pour supprimer uniquement les fichiers portant une extension particulière (par exemple : .EXT), utilisez la commande suivante :

```
del /s monDossier\*.ext
```

Notez que cette commande va supprimer tous les fichiers d'un type donné dans un dossier et tous ses sous-dossiers. Vous pouvez par exemple utiliser cette commande pour supprimer tous les fichiers temporaires portant l'extension « .TMP ».

V-B - Gestion avancée des dossiers (répertoires)

La commande « rmdir » ou sa jumelle « rd » ont été améliorées par rapport à l'ancienne version DOS. Pour supprimer des dossiers (répertoires), utilisez « rmdir » (ou simplement « rd »). Les dossiers doivent être vides pour pouvoir être supprimés. Cependant, il existe un commutateur « /s » qui permet de supprimer les sous-dossiers ainsi que les fichiers. Le comportement est similaire à l'ancienne commande « deltree ». La commande :

```
rd /s monDossier
```

va supprimer « monDossier » et tout ce qu'il contient. Il est également possible d'utiliser le commutateur « /q » pour le mode silencieux qui ne demande pas de confirmation.

Des options ont aussi été ajoutées à la commande de création de dossiers. La commande « mkdir » ou « md » peut maintenant créer une arborescence complète. Par exemple, considérons la commande suivante :

```
md nouveau\nouveau1\nouveau2
```

Celle-ci va créer un dossier « nouveau » avec un sous-dossier « nouveau1 » qui contiendra lui-même un sous-dossier « nouveau2 ».

V-C - Xcopy : Une commande puissante et polyvalente pour copier et sauvegarder les fichiers

Pour copier un grand nombre de fichiers ou pour sauvegarder un dossier volumineux, passer par la ligne de commande est souvent la solution la plus rapide et facile. La commande « xcopy » est une commande puissante et très utile qui permet d'effectuer ces opérations. (L'ancien DOS avait aussi une commande « xcopy », mais il y avait parfois des problèmes avec les noms de fichiers trop longs. La version de xcopy présente dans Windows XP ne présente plus ces problèmes).

La commande « xcopy » est fournie avec une soupe alphabétique de **commutateurs** qui lui donnent sa polyvalence et permettent son utilisation comme un utilitaire de sauvegarde de fichiers. Entrez « xcopy /? » dans une invite de commande pour voir tous les commutateurs utilisables.

Par exemple, avec l'option « /d:[jj-mm-aaaa] », seuls les fichiers ayant changé depuis cette date sont copiés. Si l'on veut par exemple copier tous les fichiers du répertoire « monDossier » qui ont été modifiés depuis le 01/06/2003 et les placer dans le répertoire « backup », on utilise la commande

```
xcopy monDossier backup /d:01-06-2003
```

Si aucune date n'est spécifiée, le commutateur /d copie tous les fichiers qui ont changé depuis n'importe quelle date. S'il y a un espace dans un nom ou un chemin, il faut encadrer le nom par des guillemets.

Ajouter le commutateur « /s » permet de copier également les sous-répertoires et leur contenu. D'autres options permettent de copier les fichiers en lecture seule et cachés. Au total, il y a environ 27 options de ligne de commande pour **xcopy**. Vous trouverez plus d'informations sur [cette page](#).

V-D - Comment lister des fichiers avec « dir »

La commande « dir » n'est pas nouvelle, mais elle reste l'une des plus utiles pour la plupart des utilisateurs de PC. Si vous avez toujours rêvé de pouvoir établir une liste des fichiers contenus dans un répertoire particulier, vous avez sans doute remarqué que l'interface graphique de Windows ne fournit aucun moyen simple pour le faire. Il est cependant très simple d'effectuer cette opération via la ligne de commande.

La commande « dir monDossier » affiche la liste des fichiers et sous-répertoires du dossier « monDossier ». Comme précédemment, plusieurs commutateurs sont utilisables pour modifier le comportement de la commande. Par exemple « /h » affichera les fichiers cachés et « /s » affichera, en plus du contenu du répertoire courant, le contenu des sous-dossiers du répertoire courant.

La sortie de la commande s'affiche sur l'écran par défaut. Pour enregistrer la sortie de la commande dans un fichier, il faut utiliser la syntaxe suivante :

```
dir monDossier >listeFichiers.txt
```

(où « > » est le symbole de redirection).

Le fichier listeFichiers.txt sera alors créé. La sortie peut aussi être dirigée directement vers l'imprimante sur le port parallèle (LPT1), mais je pense qu'il est préférable de créer un fichier puis ouvrir ce fichier pour l'imprimer. En effet, la liste peut être plus longue que ce que vous pensiez. La commande « dir » peut également être utilisée dans Windows 98/Me.

La commande « dir » peut également être utilisée pour obtenir une liste des fichiers ayant une certaine extension. Par exemple

```
dir monDossier\*.doc /s >listeFichiers.txt
```

va lister tous les documents Word (.doc) contenus dans « monDossier » et tous ses sous-dossiers. Cette syntaxe affichera également les détails sur les fichiers. Si vous voulez uniquement les noms de fichiers, utilisez le commutateur « /b ».

V-E - Renommer des fichiers avec « ren »

Renommer un grand nombre de fichiers peut être une opération fastidieuse. La commande « ren » (ou « rename ») est limitée, mais peut être utile notamment avec l'utilisation de caractères joker. La syntaxe de base est :

```
ren fichier1 fichier2
```

Le fichier renommé sera obligatoirement dans le même dossier que l'original, cette commande ne peut pas déplacer le fichier vers un autre répertoire. L'utilisation de jokers permet de changer par exemple l'extension de certains fichiers. Ainsi, la commande suivante :

```
ren *.txt *.doc
```

va changer l'extension de tous les fichiers textes (.txt) en .doc.


V-F - Déplacer des fichiers avec « move »

La commande « move » déplace un fichier d'un répertoire à un autre. La commande accepte les caractères joker, ce qui la rend utile pour déplacer tous les fichiers d'un type donné vers un autre emplacement. La commande suivante :

```
move /y dossier1\*.mp3 dossier2\
```

déplacera les fichiers MP3 du dossier 1 vers le dossier2. Le commutateur « /y » est utilisé pour que le système ne demande pas de confirmation d'écrasement. Si vous souhaitez empêcher les écrasements, utilisez l'option « /-y » .

V-G - Changer les attributs de fichier avec « attrib »

En plus de leur contenu, les fichiers disposent également d'un ensemble de propriétés qui les caractérisent (parfois appelées  **metadata**). Un exemple connu de ces propriétés est les quatre attributs : lecture seule, caché, système et archive. L'attribut archive est principalement utilisé dans les procédures de sauvegarde, mais les trois autres attributs sont utilisés dans différents contextes. Ces attributs peuvent être activés ou désactivés avec la commande « attrib » et ses options. La grande utilité de cette commande est qu'elle peut agir sur les sous-répertoires avec son commutateur « /s ». Elle peut aussi utiliser le joker « * ». Par exemple, pour enlever les attributs « lecture seule », « système » et « caché » d'un fichier, il faut utiliser la syntaxe :

```
attrib -r -s -h nom-fichier
```

Cette commande est souvent utilisée lorsqu'il faut modifier certains fichiers système. Pour restaurer les attributs, la commande est :

```
attrib +r +s +h nom-fichier
```

Une utilisation courante de cette commande est la suppression de l'attribut « lecture seule » sur des fichiers copiés à partir d'un CD-ROM. Par défaut, les fichiers provenant de CD sont marqués comme étant en « lecture seule ». Cela peut poser des problèmes si l'on souhaite les modifier. Pour enlever l'attribut « lecture seule » de tous les fichiers dans un répertoire et ses sous-répertoires, utilisez

```
attrib -r unDossier\*.*/s
```

Si vous voulez traiter uniquement les fichiers d'un certain type comme les MP3, utilisez :

```
attrib -r unDossier\*.mp3
```

V-H - Comment rendre vos dossiers favoris facilement accessibles dans les boîtes de dialogue Windows

Une autre ancienne commande que je trouve facile à utiliser est « subst ». Il y a certains dossiers que j'utilise en permanence et j'aime avoir un accès direct à ceux-ci. Une méthode pour faire cela est d'utiliser la commande « subst » pour assigner une lettre de lecteur à un dossier. Comme les lecteurs et les disques se situent dans « Poste de travail » (ou « Ordinateur » dans Vista), ils sont toujours facilement accessibles. Pour associer le lecteur « Z: » au dossier « monDossier », utilisez :

```
subst z: monDossier
```

À moins que vous ne soyez positionné dans le répertoire parent de « monDossier », vous devez utiliser le chemin complet vers le dossier. Cette association entre la lettre de lecteur est maintenue jusqu'à ce que l'utilisateur se déconnecte ou que l'ordinateur soit éteint. C'est pourquoi, je me suis créé un fichier batch d'une ligne dans mon dossier « Démarrage » (dans le menu démarrer) qui rétablit cette association.

V-I - Autres commandes

Il existe tellement de commande que je ne peux en donner ici qu'un aperçu. Une liste des commandes qui, je pense, sont les plus utiles sur les ordinateurs domestiques est disponible [sur cette page](#). Je vous invite à lire ceci attentivement, car vous serez surpris par l'utilité de certaines commandes. Par exemple, il existe toute une série de commandes pour contrôler les fonctions réseau et pour l'utilisation d'Internet. (Beaucoup de ces commandes commencent par « net... »). Il y a de nouvelles commandes, mais aussi des commandes bien connues comme « ping » et « tracert ». Comme de plus en plus de gens ont un réseau chez eux, ces commandes ont maintenant plus de signification pour la plupart des utilisateurs. Rendez-vous [sur cette page](#) pour des explications sur certaines commandes réseau.

VI - Configurer la fenêtre d'invite de commande



Ce chapitre est une traduction de la page <http://commandwindows.com/configure.htm> dont l'auteur original est Victor Laurie.

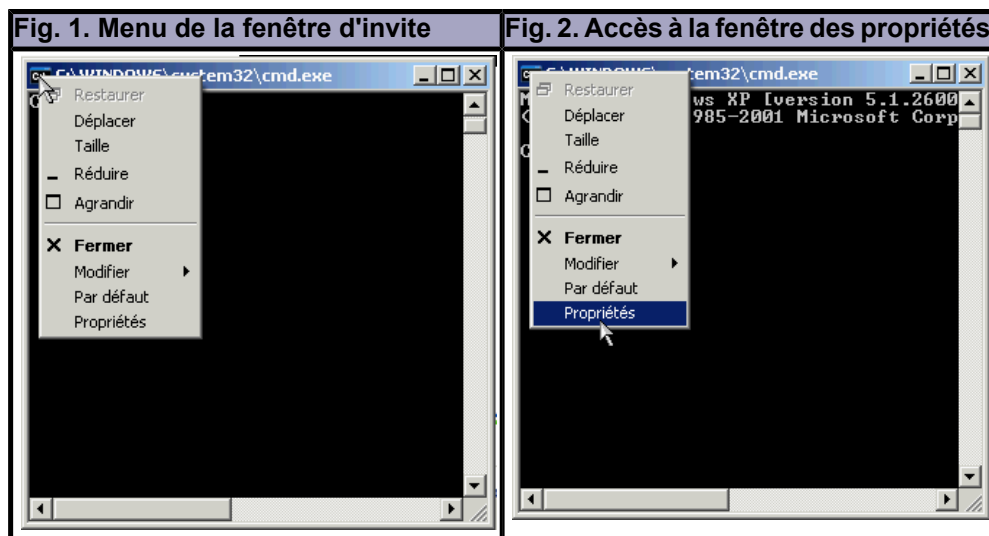
L'interface de la console de commande peut être personnalisée de plusieurs façons décrites ci-dessous.

VI-A - Introduction

Les réglages par défaut pour la fenêtre d'affichage de l'invite de commande peuvent être changés pour s'accorder avec vos préférences personnelles. Les propriétés qui peuvent être modifiées sont celles qui affectent l'apparence, telles que la taille de la fenêtre, la couleur du texte et de l'arrière-plan et la police. D'autres ont une incidence sur le fonctionnement comme le mode d'édition et la taille du buffer d'historique qui peuvent être configurés.

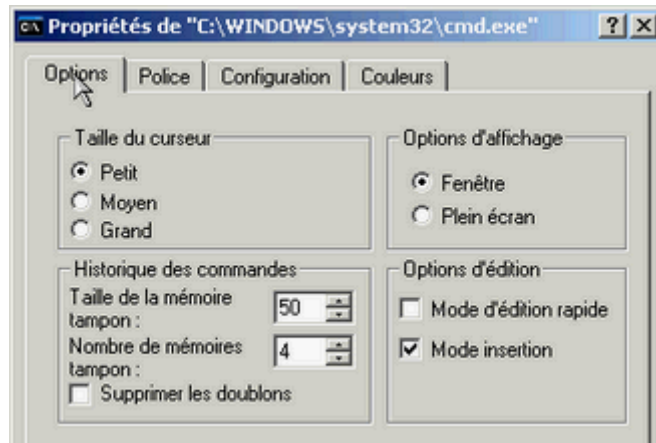
VI-B - La boîte de dialogue des propriétés de l'invite de commande

Le changement des options de l'invite de commande peut être effectué via les propriétés de cette boîte de dialogue. L'accès à celle-ci se fait par le menu que l'on obtient en cliquant (bouton gauche) sur le coin supérieur gauche de la fenêtre ou par un clic droit sur la barre de titre de la fenêtre. L'accès à celle-ci se fait par le menu que l'on obtient en cliquant (bouton gauche) sur le coin supérieur gauche de la fenêtre ou par un clic droit sur la barre de titre de la fenêtre. Les figures 1 et deux indiquent la procédure à suivre.



La fenêtre des propriétés contient quatre onglets. Chaque onglet est décrit ci-dessous.

VI-B-1 - Onglet Options



La figure ci-dessus montre l'onglet options. Les réglages sur cet onglet permettent la configuration de la taille du curseur et l'affichage en mode fenêtre ou plein écran. Le mode « Édition rapide » vous permet d'utiliser la souris pour copier et coller du texte dans l'invite de commande. Vous pouvez également choisir d'insérer ou d'écraser lors de la saisie en cochant ou décochant « Mode insertion »

VI-B-2 - Tampon d'historique des commandes

Une nouvelle fonctionnalité est le tampon « Historique des commandes ». Ce tampon enregistre les commandes que vous avez tapées précédemment afin que vous n'ayez pas à les retaper (reliques de Doskey). La valeur par défaut est 50, mais peut prendre des valeurs jusqu'à 999 (rappelez-vous que cela utilise la mémoire vive RAM). Pour naviguer dans les commandes saisies précédemment, utilisez les flèches [Haut] et [Bas] du clavier. Vous pouvez aussi appuyer sur [F7] pour afficher une fenêtre pop-up avec la liste des commandes précédentes.

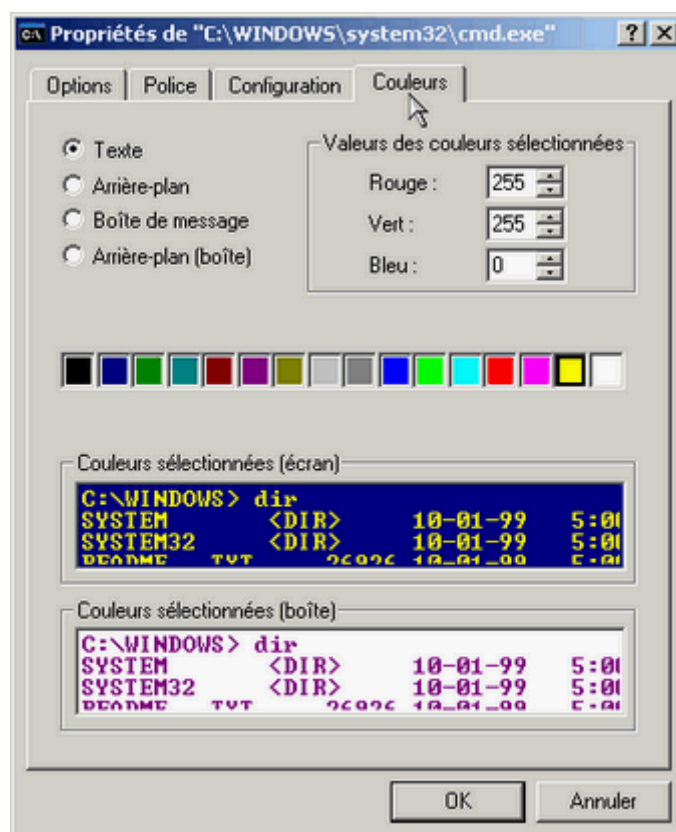
VI-B-3 - Onglet Police

Cet onglet se comprend facilement. Sur la plupart des machines, il y a le choix entre deux ou trois polices : Raster (par défaut) et Lucida Console est plus polyvalente. Il est aussi possible de choisir la taille de la police. Comme pour la plupart des réglages, vous pouvez choisir si les changements s'appliquent à la fenêtre courante uniquement ou à toutes les fenêtres ayant le même titre.

VI-B-4 - Onglet Configuration

La position et la taille de la fenêtre peuvent être paramétrées dans cet onglet. Il y a également des réglages pour la taille du tampon écran. Cette taille détermine combien de lignes qui s'inscrivent sur la console sont mémorisées. (Cette possibilité de faire défiler l'historique de toutes les commandes précédentes n'était pas présente dans DOS. Une fois que le texte avait disparu de l'écran, il n'était pas possible de faire défiler l'écran pour le retrouver).

VI-B-5 - Onglet Couleurs

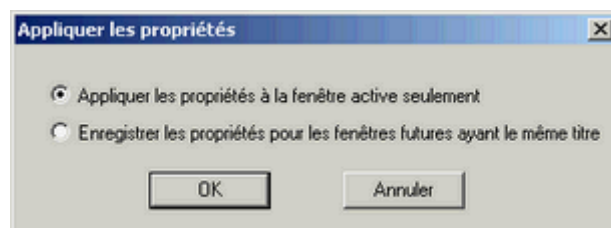


L'onglet de configuration des couleurs est présenté ci-dessus. Les couleurs par défaut pour la fenêtre d'invite sont celles de la vieille combinaison « noir et blanc ». Cependant, d'autres couleurs sont disponibles. La couleur de la fenêtre pop-up associée à la fenêtre d'invite peut également être configurée. Comme montré dans la figure ci-dessus, les couleurs peuvent être changées en cliquant sur la couleur de votre choix ou en saisissant la valeur numérique en notation RGB 256. Sur cette figure, j'ai choisi un arrière-plan bleu avec le texte en jaune. Vous pouvez choisir si les changements de couleur s'appliquent à la fenêtre courante uniquement ou à toutes les fenêtres de console.

Les couleurs de la console peuvent également être changées avec la commande « color » ou avec un commutateur d'invite de commande qui modifie directement l'interpréteur de commande. Ces deux méthodes sont décrites ci-dessous.

Il est aussi possible de créer des raccourcis qui ouvrent la ligne de commande avec plusieurs jeux de couleurs (voir le paragraphe « Raccourcis » ci-dessous).

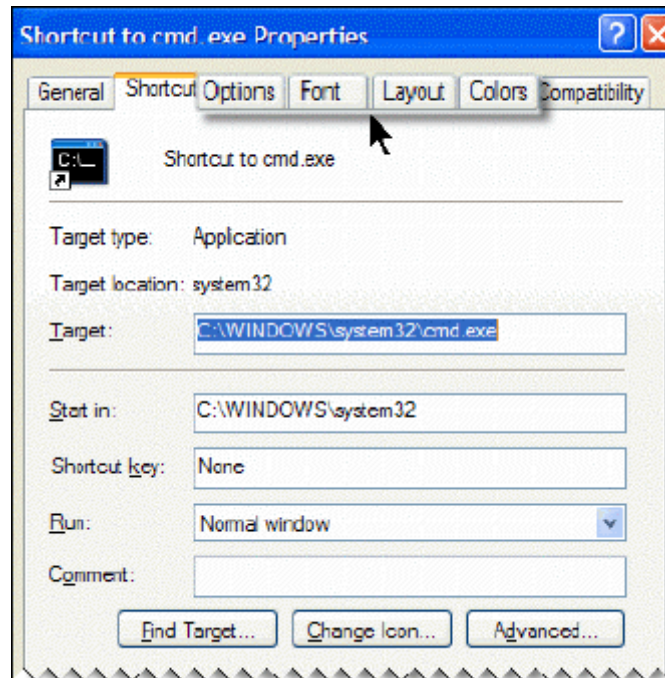
VI-C - Application des changements à toutes les fenêtres d'invite



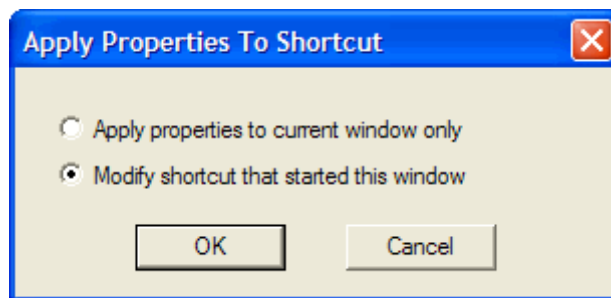
Remarquez que les changements ne s'appliquent qu'à la fenêtre courante à moins que vous ne précisiez le contraire. Lorsque vous effectuez des changements dans la configuration, vous obtiendrez la fenêtre ci-dessus. Pour que les

changements persistent après la fermeture de la console courante, cochez le bouton « Enregistrer les propriétés pour toutes les fenêtres de même nom ».

VI-D - Configurer des raccourcis pour l'invite de commande



Si vous utilisez fréquemment l'invite de commande, vous allez placer un raccourci sur le bureau ou dans la barre de lancement rapide. Après la création du raccourci, vous pouvez configurer les propriétés de la console qui sera ouverte avec la plupart des options dont nous avons parlé ci-dessus. Faites un clic droit sur le raccourci et choisissez « Propriétés » dans le menu contextuel. Quatre des onglets de la feuille de propriétés sont identiques à ceux décrits ci-dessus. Vous pouvez les remarquer dans la figure ci-dessus. Vous pouvez effectuer la personnalisation que vous souhaitez et celle-ci s'appliquera uniquement à la console ouverte par le raccourci.



Il est également possible d'effectuer la configuration à partir de la fenêtre ouverte par ce raccourci. Après configuration, la boîte de dialogue ci-dessus va apparaître. Cocher le bouton « Modifier le raccourci qui a lancé cette fenêtre ».

VI-E - La commande « color »

Les couleurs d'affichage peuvent être configurées directement dans la console à l'aide de la commande « color » suivie par un nombre à deux chiffres (en hexadécimal). Le premier chiffre détermine la couleur de l'arrière-plan et le second la couleur du texte. Le tableau ci-dessous détaille la couleur associée à chaque chiffre hexadécimal.

Tableau I. Codes couleur hexadécimaux

0 = Noir	8 = Gris
1 = Bleu	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

Par exemple la commande « color 1E » donnera un arrière-plan bleu avec un texte jaune. Les modifications de couleurs effectuées par cette commande ne s'appliquent qu'à la session courante. La commande « color » sans argument réinitialisera les couleurs aux valeurs par défaut du système.

VI-F - Options pour l'interpréteur de commande cmd.exe

Une autre manière de modifier le comportement de l'invite de commande est d'utiliser des commutateurs avec l'interpréteur de commande *cmd.exe*. La syntaxe pour *cmd.exe* est :

```
cmd [[{/c|/k}] [/s] [/q] [/d] [{/a|/u}] [/t:fg] [/e:{on|off}] [/f:{on|off}] [/v:{on|off}] chaine]
```

Le tableau II décrit brièvement les paramètres. Consultez l'aide de Windows ou le Centre de Support pour plus de détails.

Tableau II. Commutateurs pour cmd.exe

Paramètre	Description
/c	Exécute la commande donnée par la <i>chaîne</i> de caractères puis se termine.
/k	Exécute la commande donnée par la <i>chaîne</i> de caractères et reste actif.
/s	Modifie le traitement de la <i>chaîne</i> après /C ou /K. Consultez l'aide de Windows pour plus de détails
/q	Désactive l'écho local. Par défaut l'écho est actif (on)
/d	Désactive l'exécution des commandes AutoRun
/a	Redirige la sortie de commandes internes vers un canal ou un fichier ANSI. (par défaut)
/u	Redirige la sortie de commandes internes vers un canal ou un fichier UNICODE.
/t:fg	Change la couleur du premier plan <i>f</i> ou de l' <i>arrière-plan</i> . Les codes hexa pour <i>f</i> et <i>g</i> sont décrits dans le tableau I.
/e:on /e:off	Active ou désactive les extensions de commande. La valeur par défaut est « on ».
/f:on /f:off	Active ou désactive la complétion des noms de fichiers et répertoires
/v:on /v:off	Active ou désactive l'expansion retardée des variables d'environnement
<i>chaîne</i>	Spécifie la commande à exécuter

VII - L'invite Démarrer-Exécuter

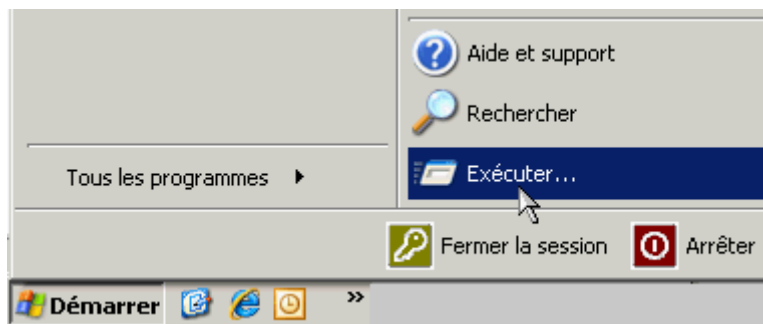


Ce chapitre est une traduction de la page <http://commandwindows.com/runline.htm> dont l'auteur original est Victor Laurie.

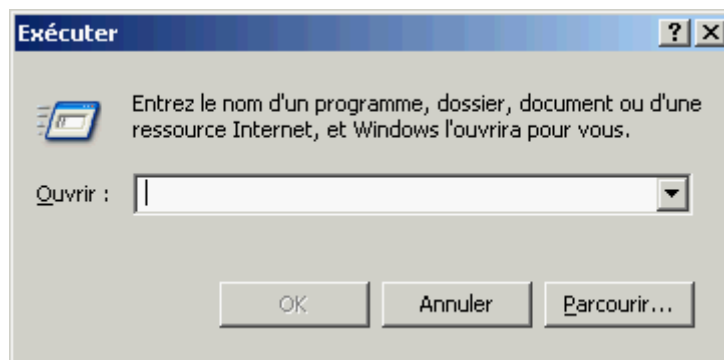
La fenêtre « exécuter » du menu Démarrer peut être utilisée pour accélérer l'accès à une grande variété de fonctions. Quelques exemples des raccourcis disponibles sont décrits ici.

VII-A - Introduction à la fenêtre « Exécuter »

La boîte de dialogue « exécuter » est peut-être la fonction la moins utilisée du menu démarrer. C'est bien dommage, car cet outil peut s'avérer très utile. C'est souvent le moyen le plus rapide pour lancer des programmes ou pour ouvrir des répertoires ou des documents. L'image ci-dessous montre l'élément **Démarrer-Exécuter**.



Lorsque vous cliquez sur « Exécuter », la boîte de dialogue ci-dessous s'affiche. Vous pouvez y saisir des commandes.



VII-B - Ouvrir des applications dans « Exécuter »

Bien que les applications puissent être ouvertes de plusieurs façons, passer par « Exécuter » est souvent le moyen le plus rapide. Les icônes de raccourcis sur le bureau sont également un moyen rapide, mais vous devez créer un raccourci pour toutes les applications et vous pouvez alors avoir des dizaines d'icônes.

Les applications les plus adaptées à l'utilisation de la ligne de commande RUN sont celles qui se trouvent dans la variable d'environnement PATH. Rendez-vous sur [cette page](#) pour plus de détails sur le « path ». La variable d'environnement « PATH » représente un ensemble de dossiers dont les noms n'auront pas besoin d'être indiqués lors de la saisie d'une commande. La variable « PATH » inclut par défaut `\\Windows\\` et `\\Windows\\system32\\`. Beaucoup d'accessoires standard de Windows se trouvent dans ces dossiers et peuvent être ouverts en entrant simplement le nom de l'exécutable. Des commandes que j'utilise couramment sont l'éditeur de Registre (regedit) et [l'utilitaire](#)

de configuration système (msconfig). Remarquez qu'aucun de ces outils système fréquemment utilisés n'a d'icône dans **Démarrer->Tous les programmes**. La fenêtre « Exécuter » est le moyen standard pour y accéder.




Vous trouverez ci-dessous une liste de quelques applications qui peuvent être lancées depuis la fenêtre « Exécuter ».

Quelques applications pour la fenêtre « Exécuter »

Commande à saisir	Fonction
calc	Ouvre la calculatrice
cmd	Ouvre la fenêtre d'invite de commande
explorer	Ouvre l'explorateur WIndows
magnify	Ouvre la loupe
msconfig	Utilitaire de configuration du système
mshearts	Ouvre la dame de pique
msinfo32	Outi d'informations Système
mspaint	Ouvre Paint
notepad	Ouvre le bloc-notes
regedit	Éditeur de Registre
sol	Ouvre le jeu du solitaire

Les programmes qui ne sont pas dans le « path » doivent être appelés par leur chemin complet (dossier racine et tous les sous-dossiers). Il n'est pas nécessaire de saisir les noms de chemins longs puisqu'un bouton « Parcourir » est disponible sur la fenêtre. C'est un moyen de rechercher les fichiers qui vous intéressent et d'y accéder directement sans avoir à taper. Si vous le souhaitez, vous pouvez ajouter les chemins des programmes que vous utilisez souvent en utilisant les méthodes décrites dans [ce paragraphe](#).

VII-B-1 - Ouvrir les consoles de gestion

Les  **consoles de gestion** sont des fonctions qu'il est pratique de lancer via la fenêtre « Exécuter ». Par exemple, le  **défragmenteur de disque** peut être ouvert en saisissant « dfrg.msc ». Je trouve cet accès beaucoup plus facile que par les autres moyens (qui nécessitent plusieurs étapes). Le tableau ci-dessous détaille une liste de nom de fichier pour ouvrir divers services. Une liste plus complète est disponible  [sur cette page](#).

Commandes pour quelques Consoles de Gestion (extension .msc obligatoire)

Commande à saisir	Fonction
<i>ciadv.msc</i>	Gère le Service d'Indexation
<i>compmgmt.msc</i>	Console de gestion de l'ordinateur. Elle contient plusieurs autres consoles.
<i>devmgmt.msc</i>	Gestionnaire de périphériques
<i>dfrg.msc</i>	Défragmenteur de disque
<i>diskmgmt.msc</i>	Gestion des disques
<i>gpedit.msc</i>	Éditeur de stratégies de groupe Uniquement dans Windows XP Professionnel
<i>services.msc</i>	Gère les nombreux services de Windows et des logiciels installés

VII-B-2 - Ouvrir les éléments du panneau de configuration

Il est également possible d'utiliser **Exécuter** pour ouvrir les éléments qui apparaissent dans le *Panneau de Configuration*. Une description complète des raccourcis du Panneau de configuration est disponible [sur cette page](#). Par exemple, « main.cpl » lance la fenêtre de propriétés de la souris.

VII-B-3 - Rundll32.exe

Il y a un certain nombre de commandes utilisant Rundll32.exe qui peuvent être entrées dans **Démarrer->Exécuter**. Une description complète peut être [consultée ici](#).

VII-C - Ouvrir des dossiers dans « Exécuter »

Les fichiers, mais aussi les dossiers peuvent être ouverts depuis la fenêtre « Exécuter ». Les dossiers concernés sont ceux inclus dans le « path ». Exemple : les dossiers contenus dans *Windows* et *Windows\system32* comme les dossiers [Font](#)s et « Drivers ». Les dossiers qui se trouvent dans *Documents and Settings\{Utilisateur}* peuvent également être ouverts via « Exécuter ». Par exemple le dossier [SendTo](#) (en un seul mot). Il est pratique d'utiliser « Exécuter » pour ouvrir ce dossier afin de le modifier. Cela permet de rajouter facilement des fonctions au menu contextuel « Envoyer vers... ». ([Voir cette page](#)). Un autre exemple est le dossier *Cookies*.
Remarque pour Windows Vista : certains dossiers système comme SendTo et Cookies ne sont pas accessibles directement dans Vista. Consultez la [page sur les commandes Shell](#).

Il y a également quelques raccourcis intéressants vers les dossiers qu'il peut être utile de connaître. Taper backslash (\) dans la fenêtre « Exécuter » et faire [Entrée] vous amènera dans le dossier racine, il s'agit en général du dossier C:. Si vous entrez un point (.), l'explorateur s'ouvrira sur le dossier *Documents and Settings\{Utilisateur Courant}* dans Windows XP (ou *Utilisateurs\{Utilisateur Courant}* dans Vista). Si vous entrez deux points (..) l'explorateur ouvrira *Document and Settings* (ou *Utilisateurs* dans Vista).

VII-D - Glisser-Déplacer des dossiers et des fichiers dans la fenêtre « exécuter ».

Si la fenêtre « Exécuter » est ouverte (et que la ligne est vide), des dossiers ou des fichiers peuvent être déposés (par Glisser-Déplacer) depuis un dossier ouvert. Le chemin complet de l'élément déposé sera alors inséré sur la ligne. Si vous appuyez sur « OK » ou tapez sur [Entrée] l'élément va s'ouvrir. Bien qu'en général cette fonctionnalité ne présente aucun avantage, elle peut être utile pour ceux qui ont des problèmes pour double-cliquer avec la souris.

VII-E - Ajouter des applications au « path »

Pour pouvoir lancer un programme simplement par son nom, il faut placer le chemin de ce programme dans le PATH. Pour savoir comment ajouter des dossiers au PATH, [regarder cette page](#)

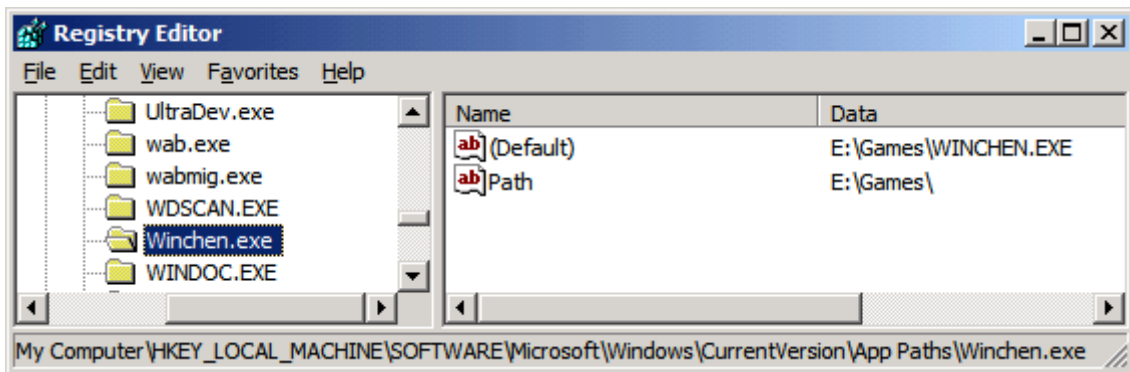
Vous pouvez aussi modifier le registre pour lui indiquer explicitement les emplacements de vos exécutables favoris. La clé qui gère cela est :

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\AppPaths
```

- 1 Créez une nouvelle sous-clé portant le nom de l'exécutable que vous voulez ajouter au path. *Par exemple : HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\AppPaths\unFichier.exe*
- 2 Dans cette nouvelle clé, ajouter une valeur chaîne appelée « Path » contenant le chemin vers l'exécutable. *Par exemple : C:\Program files\repertoireDuProgramme*
- 3 La nouvelle clé contiendra par défaut une variable vide (Default). Éditez-la pour y placer l'emplacement complet du programme exécutable. *Par exemple : C:\Program files\repertoireDuProgramme\unFichier.exe*

Vous pouvez maintenant entrer directement « unFichier.exe » dans la fenêtre « Exécuter » pour lancer le programme. Vous pouvez voir un exemple de clés du registre dans l'image ci-dessous. J'ai ajouté un jeu d'échecs avec l'exécutable « winchen.exe » dans le PATH.

Un certain nombre de programmes sont normalement déjà présents dans votre système. Beaucoup d'applications s'y insèrent elles-mêmes lorsqu'elles s'installent. Les composants de Microsoft Office en sont un bon exemple. Il suffit par exemple de saisir « winword » dans « Exécuter » pour ouvrir Microsoft Word ou « excel » pour lancer Microsoft Excel.



VII-F - Applications Internet

Si vous êtes connecté à Internet et que vous entrez une URL dans la fenêtre « Exécuter », Internet Explorer (ou le navigateur par défaut) s'ouvrira sur le site web choisi. Le bouton « Parcourir » peut être utilisé pour naviguer dans le dossier « Favoris ». Vous pouvez alors cliquer sur un lien. (Assurez vous que la boîte de dialogue « Parcourir » a bien le type de fichier « Tous les fichiers » sélectionné. Vous pouvez aussi écrire un e-mail en entrant « mailto:quelquun@quelquepart.com ». Cela va ouvrir l'éditeur d'e-mail avec l'adresse que vous avez entrée dans le champ destinataire. Si vous utilisez un autre client e-mail que Microsoft Outlook Express, celui-ci doit être associé avec la fonction « mailto ». La plupart des clients e-mail le font automatiquement lorsqu'ils sont installés. (D'après ce que je sais, cela ne marche pas avec AOL).

Une recherche sur Google est possible grâce à une commande simple. Saisissez « www.google.com/search?q=%1 » dans la fenêtre « Exécuter », où %1 est le terme que vous recherchez. Vous pouvez rechercher plusieurs mots en utilisant le signe plus (+). Essayez par exemple « www.google.com/search?q=registre+windows » (sans les guillemets).

VII-G - Comparaison entre la fenêtre « Exécuter » et l'invite de commande

Bien qu'une grande partie des commande puisse être utilisée à la fois dans l'invite de commande et dans la fenêtre « Exécuter », certaines d'entre elles ne fonctionnent que dans l'une ou dans l'autre. Les commandes intégrées à l'interpréteur de commande ne peuvent pas être appelées dans la fenêtre « Exécuter » sans appeler auparavant *cmd.exe*. Ces commandes sont [listées ici](#). Elles incluent les commandes comme « dir » et « del ». Certaines fonctions spéciales de la fenêtre « Exécuter » comme l'ouverture directe de dossiers ou de raccourcis Internet (décrites précédemment) ne fonctionnent pas dans l'invite de commande (à moins d'être précédées de la [commande « start »](#)).

VII-H - Accéder à la console de commande depuis la fenêtre « Exécuter »

L'interpréteur de commande peut être appelé depuis la fenêtre « Exécuter » en entrant

```
cmd /c une_commande
```

Avec le commutateur « /c », *une_commande* va être lancée puis la console de commande va se fermer. Si vous souhaitez que la console reste ouverte, utilisez le commutateur « /k ». Exemple :


```
cmd /k une_commande
```

VII-I - La fenêtre « Exécuter » dans Vista

Le menu *Démarrer* de Vista n'affiche pas l'élément *Exécuter* par défaut. La plupart des fonctions de *Exécuter* peuvent être utilisées dans la nouvelle barre « Rechercher » en bas du menu *Démarrer* de Vista, mais j'aime encore utiliser « Exécuter » de temps en temps. Vous pouvez ouvrir la fenêtre *Exécuter* temporairement en utilisant le raccourci clavier **[Windows] + [R]** Pour remettre *Exécuter* dans le menu *Démarrer* :

- 1 Faites un clic droit sur le menu *Démarrer* et cliquez sur « Propriétés »
- 2 Sélectionnez l'onglet « Menu Démarrer » puis cliquez sur « Personnaliser... »
- 3 Cochez la case « Commande Exécuter »

VIII - Gestion des fichiers à partir de la ligne de commande - Assoc et Ftype



Ce chapitre est une traduction de la page <http://commandwindows.com/assoc.htm> dont l'auteur original est Victor Laurie.

Windows est fourni avec plusieurs outils en ligne de commande permettant de gérer les fichiers. Les fonctionnalités et applications de Assoc et Ftype sont décrites ici.

VIII-A - Introduction

Avant de commencer l'explication des outils, je vais vous expliquer rapidement quelques bases sur la manière dont Windows gère les fichiers. (Vous trouverez plus de détails sur [cet autre site](#)). L'une des propriétés les plus basiques d'un fichier est son *type*. Chaque type de fichier est associé à un ensemble d'actions qui peuvent lui être appliquées. Le logiciel qui est sélectionné pour effectuer ces actions avec les fichiers d'un certain type est dit « associé » au type de fichier. Il peut y avoir plusieurs actions possibles et différents logiciels peuvent être utilisés pour chaque action. Cet ensemble de logiciels constituent les *associations de programmes* pour un type de fichier donné. L'*extension* d'un fichier indique à l'ordinateur de quel type est le fichier et ce qui doit être fait quand le fichier est ouvert, double-cliqué ou invoqué d'une manière ou d'une autre. Microsoft utilise également le mot « associé » pour désigner l'association entre un type de fichier particulier et une extension. Toutes ces informations sont stockées dans la base de registre et peuvent être modifiées de plusieurs façons. Les méthodes utilisant l'interface graphique sont décrites sur [un autre site](#). Nous allons voir ici comment faire cela en ligne de commande.

VIII-B - Gestion des types de fichiers et des extensions avec la commande « assoc »

Cet outil est très utile pour gérer les relations ou associations entre les extensions et les types de fichiers. La syntaxe est :

```
assoc [.ext[=[Type du fichier]]]
```

Si vous saisissez la commande « assoc » seule, vous obtiendrez la liste des types de fichiers associés aux extensions enregistrées sur le système. La liste peut être longue donc il vaut mieux rediriger la sortie vers un fichier ou utiliser la commande « more » pour afficher une page après l'autre :

```
assoc > liste.txt
ou
assoc | more
```

Par exemple, pour afficher le type de fichier associé à l'extension *.txt*, tapez la commande suivante :

```
assoc .txt
```


Pour supprimer l'association entre un type de fichier et une extension *.xyz* (à utiliser avec précaution), entrez :

```
assoc .xyz=
```

Une autre utilisation est d'associer une extension donnée avec un certain type de fichier. Par exemple, pour associer l'extension *.log* avec le type *txtfile*, utilisez

```
assoc .log=txtfile
```

Remarquez que plusieurs extensions peuvent être associées à un type de fichier. Par exemple, le type de fichier « jpegfile » est associé aux deux extensions `.jpg` et `.jpe`. Sachez aussi qu'il est possible de créer sa propre extension de fichier et de l'associer à un type de fichier.

Pour une référence sur `assoc` consultez  [ceci](#) ou le centre d'aide et de support de Windows. Vous pouvez aussi utiliser la commande :

```
assoc /?
```

VIII-C - Gérer les types de fichiers et les associations de programmes avec la commande « `ftype` »

Comme je l'ai mentionné précédemment, à chaque type de fichiers est associé un ensemble d'opérations et de logiciels correspondants. En particulier, tous les types de fichiers actifs possèdent une action nommée « Open » qui est l'action par défaut. Il s'agit de l'action qui est appelée quand vous double-cliquez sur un fichier possédant l'extension associée au type de fichier. Il peut y avoir d'autres actions (listées dans le menu contextuel), mais `ftype` gère uniquement « Open ». L'action Open est définie par une chaîne de caractères qui indique le chemin complet vers le fichier exécutable qui va gérer le fichier et tous les paramètres qui doivent être passés à cet exécutable. La syntaxe de `ftype` est :

```
ftype [typeFichier[=[commandeOuverture]]]
```

En utilisant la commande `ftype` seule, vous obtiendrez la liste des types ayant une commande d'ouverture. Celle-ci sera également affichée. Cette liste peut être très longue donc il vaut mieux rediriger la sortie vers un fichier ou utiliser un « pipe » vers la commande « more ». Cette liste peut être particulièrement utile pour savoir quels sont les programmes utilisés pour ouvrir les différents fichiers.

Si un type de fichier est spécifié, alors la commande pour ce type sera affichée. Prenons par exemple le type de fichier « `txtfile` » (fichier texte), si vous saisissez :

```
ftype txtfile
```

vous obtiendrez en retour

```
txtfile=%SystemRoot%\system32\NOTEPAD.EXE %1
```

Cela indique que le fichier exécutable qui ouvre les fichiers textes est `notepad.exe` situé dans le répertoire **Windows\system32** (La variable d'environnement `%SystemRoot%` permet d'indiquer le répertoire de Windows). Remarquez la présence du paramètre `%1`. Ce paramètre est nécessaire, car la commande d'ouverture a besoin du nom du fichier à ouvrir. Le paramètre `%1` représente le nom du fichier. Cette commande peut être utile quand vous voulez savoir quel programme ouvre un type de fichier particulier.

Si vous désirez changer la *commande d'ouverture* pour utiliser Wordpad à la place du bloc-notes, la commande sera :

```
ftype txtfile="G:\Program Files\Windows NT\Accessories\wordpad.exe" "%1"
```

(l'emplacement de Wordpad peut être différent d'un ordinateur à un autre. Il s'agit ici d'un simple exemple d'illustration). Remarquez que j'utilise des guillemets pour entourer le chemin, car il contient des espaces. Bien qu'il soit plus facile de changer les associations de programme par l'explorateur Windows via **Outils-Options des dossiers**, la méthode utilisant la ligne de commande que je viens de vous décrire peut également être utile, en particulier dans les fichiers batch.

IX - Xcopy et ses applications



Ce chapitre est une traduction de la page  <http://commandwindows.com/xcopy.htm> dont l'auteur original est Victor Laurie.

Ce paragraphe détaille la syntaxe et l'utilisation de la commande « Xcopy ».

De toutes les commandes exécutables en ligne de commande, Xcopy est l'une des plus utiles pour la plupart des utilisateurs de PC domestiques. Elle fournit une méthode puissante pour copier et sauvegarder des fichiers et des répertoires.

IX-A - Syntaxe de la commande Xcopy

Xcopy possède un grand nombre d'options ce qui rend cette commande très flexible. Voici la syntaxe de la commande :

```
XCOPY source [destination] [/A | /M] [/D[:date]] [/P] [/S [/E]] [/V] [/W] [/C] [/I] [/Q] [/F] [/L] [/G] [/H] [/R] [/T] [/U] [/K] [/N] [/O] [/X] [/Y] [/Y] [/Z] [/EXCLUDE:fichier1[+fichier2][+fichier3]...]
```

Les options sont indiquées en majuscules, mais la casse n'est pas importante. Vous trouverez une description des différents paramètres dans le tableau I. Dans Windows Vista, il y a un commutateur supplémentaire : « /B ». Cette option permet de copier un lien symbolique lui-même plutôt que la cible du lien.

Tableau I. Descriptions des options de la commande Xcopy

Commutateur	Description
/A	Copie uniquement les fichiers possédant l'attribut archive, ne change pas l'attribut.
/M	Copie uniquement les fichiers possédant l'attribut archive, désactive l'attribut archive. Utile pour les sauvegardes.
/D:j-m-a	Copie les fichiers modifiés à partir de la date spécifiée. Si aucune date n'est donnée, copie uniquement les fichiers dont l'heure source est plus récente que

	l'heure de destination. Utile pour les sauvegardes.
/P	Avertissement avant la création de chaque fichier de destination.
/S	Copie les répertoires et sous-répertoires à l'exception des répertoires vides.
/E	Copie les répertoires et sous-répertoires, y compris les répertoires vides. Identique à /S /E. Peut être utilisé pour modifier /T
/V	Vérifie chaque nouveau fichier. Non utilisé par Windows XP.
/W	Vous demande d'appuyer sur une touche avant la copie.
/C	Continue la copie même si des erreurs se produisent.
/I	Si la destination n'existe pas et que plus d'un fichier est copié, considère la destination comme devant être un répertoire.
/Q	N'affiche pas les noms de fichiers lors de la copie.
/F	Affiche les noms de source et de destination complets lors de la copie.
/L	Affiche les fichiers devant être copiés.
/G	Permet la copie des fichiers chiffrés vers des destinations qui ne prennent pas en charge le chiffrement.
/H	Copie également les fichiers cachés et les fichiers système.
/R	Remplace les fichiers en lecture seule.
/T	Crée la structure de répertoires, mais ne copie pas les fichiers. N'inclut pas les répertoires ou sous-répertoires vides. /T /E incluent les répertoires et sous-répertoires vides.
/U	Copie seulement les fichiers existants déjà dans la destination.
/K	Copie les attributs. La commande normale Xcopy rétablira les attributs de lecture seule.
/N	Copie en utilisant les noms courts générés. Peut être nécessaire pour les copies de NTFS vers FAT16.
/O	Copie les fichiers d'appartenance et les informations d'ACL.
/X	Copie les paramètres d'audit de fichiers (implique /O).
/Y	Supprime la demande de confirmation de remplacement de fichiers de destination

	existants. Peut être pré-réglé dans la variable d'environnement COPYCMD.
/-Y	Provoque la demande de confirmation de remplacement d'un fichier de destination existant.
/Z	Copie les fichiers du réseau en mode redémarrable.
/EXCLUDE:fichier1[+fichier2][+fichier3] ...	Spécifie une liste de fichiers contenant des chaînes à exclure. Délicat à utiliser. Voir ci-dessous.

IX-B - Applications de Xcopy

Xcopy fournit un excellent outil pour sauvegarder les dossiers sélectionnés. Avec les options appropriées, une multitude de scénarios de sauvegarde peuvent être créés. Il est par exemple possible de configurer un script de sauvegarde copiant uniquement les fichiers qui ont été modifiés. Voici un exemple de commande :

```
xcopy C:\unDossier E:\dossierSauvegarde /D /E /C /R /H /I /K /Y
```

Cette commande va copier les fichiers cachés et en lecture seule et créer le dossier de destination et les sous-dossiers s'ils n'existent pas.

L'exemple suivant montre comment utiliser un joker; il permet de sélectionner des fichiers d'un certain type et de les copier dans un emplacement. La commande

```
xcopy C:\*.mp3 E:\dossierMP3 /S /I /C > E:\listeMP3.txt
```

va récupérer tous les MP3 du disque C: et les copier dans le répertoire dossierMP3 du disque E:. Elle crée également une liste des fichiers copiés et l'enregistre sur le disque E: (E:\listeMP3.txt). Cette simple commande préserve la structure des sous-répertoires. Si vous le souhaitez, un script plus complexe peut être écrit pour placer tous les fichiers MP3 dans un dossier sans aucun sous-répertoire.

IX-C - Exclure des fichiers et dossiers.

Une fonctionnalité utile de Xcopy est la possibilité d'exclure certains fichiers et/ou répertoires lors de la copie à l'aide du commutateur « /Exclude ». Tous les noms contenant une chaîne de caractères donnée peuvent être exclus. Malheureusement, **la description de Microsoft** sur cette option n'est pas un modèle de clarté ni d'exactitude. En fait, les discussions sur Internet à ce sujet indiquent qu'il existe une confusion considérable sur la manière d'utiliser cette fonction. J'espère que la liste ci-dessous vous aidera à mieux comprendre comment l'utiliser.

- Les chaînes de caractères pour l'exclusion ne sont pas entrées directement en paramètre de la commande.
- L'entrée attendue par le commutateur est un ou plusieurs fichiers textes qui listent les chaînes de caractères à exclure.
- Le format de l'option est : `/Exclude:{chemin}liste_des_exclusions.txt`. Remarquez la présence du deux-points (:) entre Exclude et le nom du fichier. Bien que vous puissiez utiliser plusieurs fichiers pour l'exclusion, il est préférable de n'en utiliser qu'un si possible.
- Les noms de fichiers ne peuvent pas contenir d'espaces. Il n'est pas possible d'utiliser des guillemets.
- Soyez prudents avec les chemins. Rappelez-vous que les opérations sont effectuées à partir du répertoire courant dans l'invite de commande. Je vous conseille de placer la liste d'exclusion dans le répertoire le plus haut des répertoires à copier et d'ouvrir l'invite de commande dans le répertoire juste au-dessus de celui à copier. (**Rendez-vous ici** si vous ne savez pas comment ouvrir l'invite de commande à un emplacement précis). Les différents chemins relatifs seront alors simples. Si vous utilisez l'invite de commande depuis l'emplacement par défaut, cela risque d'être compliqué.
- La liste d'exclusion doit contenir une chaîne de caractères d'exclusion par ligne. Les jokers ne peuvent pas être utilisés et ne fonctionnent pas. Si vous souhaitez exclure tous les fichiers dont le nom contient ABC,

placez simplement ABC dans une ligne du fichier d'exclusion. Si vous utilisez le commutateur « /S », le fichier d'exclusion s'appliquera aux sous-répertoires ainsi qu'au répertoire principal.

- Pour exclure un sous-répertoire particulier, indiquez son nom dans le fichier d'exclusion avec un backslash avant et après son nom. *Par exemple* : `\dossier_a_exclure\`
- Utilisez un éditeur de texte pour créer le fichier d'exclusion. N'utilisez pas Microsoft Word ou un autre traitement de texte à moins que vous ne fassiez attention à bien créer des fichiers textes purs. Utilisez l'encodage ANSI. Quand j'ai essayé d'utiliser UTF-8, le fichier n'a pas fonctionné.

IX-D - Codes retour de Xcopy

Beaucoup de commandes renvoient un code en fin d'exécution pour indiquer le statut de la commande. La liste des codes de sortie pour Xcopy est donnée dans le tableau II. Les codes d'erreurs peuvent être utiles dans les fichiers batch quand ils sont utilisés avec des **instructions if**.

Tableau II. Codes retour de Xcopy

Code retour	Description
0	Les fichiers ont été copiés sans erreur.
1	Aucun fichier à copier n'a été trouvé.
2	L'utilisateur a utilisé [Ctrl]+[C] pour arrêter xcopy .
4	Erreurs diverses comme manque de mémoire, espace disque insuffisant, nom de lecteur invalide ou syntaxe incorrecte.
5	Une erreur d'écriture disque est survenue.

X - Les commandes réseau « net » dans Windows XP



Ce chapitre est une traduction de la page <http://commandwindows.com/netservices.htm> dont l'auteur original est Victor Laurie.

Un grand nombre de commandes commençant par « net » est disponible dans Windows XP Professionnel. Ces commandes sont listées dans ce chapitre.

X-A - Les commandes « net »

La suite de commandes réseau « net » est décrite dans *l'aide de Windows et le Centre de Support*. Utilisez « net services » pour obtenir la liste des services et leur syntaxe d'utilisation. Vous pouvez aussi voir la liste des commandes « net » disponibles en tapant « net /? » dans l'invite de commande. Pour de l'aide sur la syntaxe d'une commande, tapez « net help {commande} ». Par exemple, pour de l'aide sur la commande »net stop", tapez

```
net help stop
```

La liste des commandes et une brève description de chacune d'entre elles sont données ci-dessous. Certaines de ces commandes sont des doublons de fonctions accessibles d'autres manières par exemple la [suite Netsh](#) ou le [contrôleur de services](#).

- *Net accounts* - Met à jour la base de données des comptes utilisateurs et modifie les exigences pour le mot de passe pour tous les comptes.
- *Net computer* - Ajoute ou supprime des ordinateurs de la base de données du domaine.
- *Net config* - affiche une liste des services configurables.
- *Net continue* - Continue un service qui a été suspendu par *net pause*.
- *Net file* - Affiche les noms de tous les fichiers partagés ouverts sur un serveur.
- *Net group* - Ajoute, affiche ou modifie des groupes globaux dans un domaine.
- *Net help* - Fournit une liste des commandes réseau et les sujets sur lesquels vous pouvez obtenir de l'aide.
- *Net helpmsg* - Explique pourquoi une erreur est survenue et fournit des informations de résolution des problèmes.
- *Net localgroup* - Ajoute, affiche ou modifie les groupes locaux.
- *Net name* - Ajoute ou supprime un nom de messagerie réseau.
- *Net pause* - Met en pause un service en cours d'exécution.
- *Net print* - Affiche des informations sur une file d'impressions spécifiée, affiche des informations sur toutes les files d'impressions hébergées par un serveur d'impression spécifié, affiche des informations sur un travail d'impression spécifié ou contrôle un travail d'impression spécifié.
- *Net send* - Envoie un message à un ou plusieurs utilisateurs sur le réseau.
- *Net session* - Liste ou déconnecte des sessions.
- *Net share* - Affiche ou gère les imprimantes et répertoires partagés.
- *Net start* - Liste ou démarre les services réseau.
- *Net statistics* - Affiche des statistiques pour le service Station de Travail ou Serveur Local.
- *Net stop* - Arrête des services.
- *Net time* - Affiche ou synchronise le temps réseau.
- *Net use* - Affiche ou gère les connexions à des systèmes distants.
- *Net user* - Crée des comptes d'utilisateurs locaux.
- *Net view* - Affiche les ressources ou ordinateurs du réseau.

X-B - Commande Net User dans Windows Vista

Les comptes d'utilisateurs jouent un rôle important dans Vista et « net user » fournit une méthode pour les gérer. Bien que les comptes d'utilisateurs peuvent être gérés de plusieurs manières via l'interface graphique, la ligne de commande peut être plus rapide et peut être utilisée dans les scripts.

XI - Netstat



Ce chapitre est une traduction de la page <http://commandwindows.com/netstat.htm> dont l'auteur original est Victor Laurie.

Netstat est un outil utile pour vérifier les connexions réseau et Internet. Plusieurs utilisations pour les ordinateurs personnels sont décrites dans ce chapitre, notamment la détection de connexions malveillantes.

XI-A - Syntaxe et options

La syntaxe de la commande est :

```
netstat [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [intervalle]
```

Une brève description des options est donnée dans le tableau I ci-dessous. *Remarquez que les commutateurs de Netstat utilisent le symbole « - » au lieu du slash « / »..*

Tableau I. Options de la commande Netstat

Commutateur	Description
-a	Affiche toutes les connexions et les ports en écoute
-b	Affiche l'exécutable impliqué dans la création de chaque connexion ou port d'écoute. (Ajouté dans XP SP2).
-e	Affiche les statistiques Ethernet
-n	Affiche les adresses et les numéros de ports au format numérique
-o	Affiche l'identificateur du processus propriétaire associé à chaque connexion.
-p proto	Affiche les connexions pour le protocole spécifié par proto ; proto peut être une des valeurs suivantes : TCP, UDP, TCPv6 ou UDPv6.
-r	Affiche la table de routage.
-s	Affiche les statistiques par protocole.
-v	Quand -v est utilisé avec -b, netstat va afficher la séquence des composants impliqués dans la création de la connexion ou du port d'écoute pour tous les exécutables.
[intervalle]	Affiche régulièrement les statistiques sélectionnées, en faisant une pause pendant le nombre de secondes spécifié par l'intervalle entre chaque affichage. Appuyez sur CTRL+C pour arrêter l'affichage des statistiques. Si l'intervalle est omis, netstat n'affichera les informations de configuration actuelle qu'une seule fois.

XI-B - Applications de Netstat

Netstat est l'un des nombreux outils en ligne de commande permettant de vérifier le fonctionnement d'un réseau. ([Consultez cette page](#) pour plus d'informations sur les autres outils). Elle permet de vérifier si plusieurs éléments de TCP/IP fonctionnent et quelles sont les connexions établies. À partir de Windows XP SP2, une nouvelle option « -B » a été ajoutée. Elle permet de connaître le fichier exécutable qui a ouvert une connexion. Cette nouvelle possibilité fournit un moyen de découvrir si un fichier malveillant utilise votre ordinateur de manière incorrecte en accédant à Internet. Il y a de nombreuses façons pour un administrateur système d'utiliser les options de Netstat, mais je vais vous donner deux exemples qui seront utiles à la majorité des utilisateurs de PC.

XI-B-1 - Vérifier les connexions TCP/IP

Les connexions TCP et UDP ainsi que leurs adresses IP et port peuvent être obtenues en entrant la commande :

```
netstat -an
```

La figure 1 montre un exemple de résultat obtenu.

Figure 1. Exemple de résultat pour la commande « netstat -an »

```
C:\Documents and Settings\Owner>netstat -an

Active Connections

 Proto Local Address           Foreign Address         State
 TCP   0.0.0.0:135              0.0.0.0:0               LISTENING
 TCP   0.0.0.0:445              0.0.0.0:0               LISTENING
 TCP   127.0.0.1:1027           0.0.0.0:0               LISTENING
 TCP   192.168.1.100:139        0.0.0.0:0               LISTENING
 TCP   192.168.1.100:2558      207.68.172.236:80       CLOSE_WAIT
 TCP   192.168.1.100:2916      204.14.90.25:21         CLOSE_WAIT
 TCP   192.168.1.100:2923      69.65.109.55:80         TIME_WAIT
 TCP   192.168.1.100:2924      204.245.162.25:80       ESTABLISHED
 TCP   192.168.1.100:2925      66.150.96.119:80        ESTABLISHED
 TCP   192.168.1.100:2930      204.245.162.27:80       ESTABLISHED
 UDP   0.0.0.0:445              *:
 UDP   0.0.0.0:500              *:
 UDP   0.0.0.0:1030             *:
 UDP   0.0.0.0:1040             *:
 UDP   0.0.0.0:1155             *:
 UDP   0.0.0.0:1175             *:
 UDP   0.0.0.0:4500             *:
 UDP   127.0.0.1:123            *:
 UDP   127.0.0.1:1036           *:
 UDP   127.0.0.1:1900           *:
 UDP   127.0.0.1:2922           *:
 UDP   192.168.1.100:123        *:
 UDP   192.168.1.100:137        *:
 UDP   192.168.1.100:138        *:
 UDP   192.168.1.100:1900       *:

```

Les informations affichées sont le protocole, l'adresse locale, l'adresse distante et l'état de la connexion. Remarquez que de nombreuses adresses IP sont affichées avec le numéro de port. Les différents statuts de connexion sont décrits dans le tableau II.

Tableau II. Description des différents statuts de connexion

État	Description
CLOSED	Indique que le serveur a reçu un signal ACK envoyé par le client et que la connexion est fermée.
CLOSE_WAIT	Indique que le serveur a reçu le premier signal FIN envoyé par le client et que la connexion est en train d'être fermée.
ESTABLISHED	Indique que le serveur a reçu le signal SYN envoyé par le client et que la session est établie.
FIN_WAIT_1	Indique que la connexion est toujours active, mais n'est pas utilisée actuellement.
FIN_WAIT_2	Indique que le client vient juste de recevoir l'accusé de réception du premier signal FIN envoyé par le serveur.
LAST_ACK	Indique que le serveur est en train d'envoyer son propre signal FIN.
LISTENING	Indique que le serveur est prêt à accepter une connexion.
SYN_RECEIVED	Indique que le serveur vient de recevoir un signal SYN envoyé par le client.
SYN_SEND	Indique que la connexion est ouverte et active.
TIME_WAIT	Indique que le client reconnaît la connexion comme encore activée, mais non utilisée actuellement.

XI-B-2 - Rechercher les programmes malveillants en regardant quels programmes ouvrent des connexions

Pour trouver quels sont les programmes qui établissent des connexions vers le monde extérieur, on peut utiliser la commande

```
netstat -b
```

Il est préférable d'effectuer cette vérification sur une certaine période de temps. Nous ajoutons donc un chiffre qui va indiquer à la commande de s'exécuter à intervalles fixes. Il est également préférable de conserver une trace écrite des connexions qui sont établies durant une certaine période de temps. La commande sera donc

```
netstat -b 5 >> C:\connexions.txt
```

Avec cette syntaxe, la commande va s'exécuter toutes les 5 secondes jusqu'à ce que l'utilisateur tape « Ctrl+C » (qui est la commande usuelle pour quitter). (Certains disent que cela peut entraîner une utilisation intensive du processeur de sorte que cela peut provoquer un ralentissement sur des machines « simple cœur » peu rapides. Mais cela n'était pas visible sur ma machine dual-core). La Figure 2 montre un exemple simple de résultat. Remarquez que le numéro de processus (PID) est affiché. Cette commande peut être combinée avec d'autres outils comme [le gestionnaire de tâches](#) pour analyser quels fichiers et processus sont actifs et essaient de se connecter à Internet.

Figure 2. Exemple de résultat pour la commande « netstat -b »

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	192.168.1.100:2924	204.245.162.25:80	ESTABLISHED	2104
[msfeedssync.exe]				
TCP	192.168.1.100:2558	207.68.172.236:80	CLOSE_WAIT	1684
c:\windows\system32\WS2_32.dll				
C:\WINDOWS\system32\WININET.dll				
[svchost.exe]				
TCP	192.168.1.100:2916	204.14.90.25:21	CLOSE_WAIT	2144
[Dreamweaver.exe]				

XI-B-3 - Programme batch pour vérifier les connexions et s'arrêter automatiquement

L'exemple précédent (contrôle des connexions avec « netstat -b ») présente l'inconvénient de nécessiter un arrêt manuel (CTRL+C). Il est également possible d'utiliser un fichier batch qui s'exécute un certain nombre de fois à un intervalle donné et se termine automatiquement. Dans Windows XP on peut utiliser la commande « Sleep » provenant des **outils Windows 2003 Serveur**. Voici un exemple de fichier batch :

```
@echo off
for /L %%X in (1,1,100) do (netstat -b >> C:\connexions.txt) & (sleep 5)
```

Cet exemple exécute 100 fois la commande *netstat* à 5 secondes d'intervalle.

XII - Netsh, la console des services Réseau



Ce chapitre est une traduction de la page <http://commandwindows.com/netsh.htm> dont l'auteur original est Victor Laurie.

Ce chapitre décrit la suite d'outils réseau en ligne de commande Netsh, avec sa propre console, qui est intégrée dans la plupart des systèmes Windows.

XII-A - Introduction à Netsh

Comme de plus en plus d'utilisateurs configurent un réseau chez eux, ils se transforment *de fait* en administrateurs système. Les réseaux domestiques sont très bien, mais ils nécessitent une certaine attention. Heureusement, Windows XP est fourni avec un grand assortiment d'outils en ligne de commande qui aident à maintenir votre réseau. Bien que certaines commandes soient très spécialisées et n'intéressent que les administrateurs de réseaux de taille importante, quelques outils peuvent être utiles pour les utilisateurs standards.

Certains peuvent se satisfaire des outils basiques comme *ping*, *ipconfig* et *netstat* (décrits sur [une autre page](#)). Mais les utilisateurs plus aventureux pourront se pencher sur une suite complète d'outils puissants appelée *Netsh*. Cette suite est appelée depuis l'invite de commande standard, mais possède sa propre interface (ou console) avec un grand nombre de sous-commandes. Je vais me concentrer ici sur les fonctionnalités de Netsh qui, je pense, peuvent être utiles à la plupart des utilisateurs. La suite netsh a de nombreuses applications et ceux qui souhaitent obtenir plus de détails peuvent consulter cette [référence Microsoft](#)

On accède à la console de services réseau en entrant « netsh » dans l'invite de commande. La console possède une structure hiérarchique avec des sous-console que Microsoft appelle « contextes ». Du point de vue de l'utilisateur, cela veut simplement dire que les commandes sont entrées sous la forme d'une séquence de mots. La section suivante décrit les « contextes » les plus utiles pour les utilisateurs standard.

XII-B - Le contexte « netsh diag »

Le contexte de diagnostic « diag » contient des outils utiles pour vérifier un réseau et tester différents composants et fonctions. Le tableau I liste les contextes et les sous-commandes qui présentent le plus d'intérêt. Une liste complète et détaillée est donnée dans la [référence Microsoft](#).

Tableau I. Quelques sous-console (contextes) et commandes de Netsh diag

Contexte	Sous-contexte	Commandes
diag	<i>connect</i> - Etablit, vérifie puis détruit une connexion.	<i>iphost, mail, news</i>
	<i>ping</i> - Vérifie la connectivité	<i>adapter, iphost, mail, news</i>
	<i>show</i> - Liste les composants réseau et les réglages	<i>all, client, ip, mail, modem</i>
	<i>gui</i> - Démarre l'outil de diagnostic réseau du Centre d'Aide et de Support.	Interface graphique

Cet ensemble de commandes fournit les moyens de tester quelques-unes des fonctions les plus intéressantes pour la plupart des utilisateurs. Par exemple, vous pouvez tester si votre client e-mail fonctionne ou e-mail vos réglages e-mail avec la commande

```
netsh diag connect mail
```

(Cela peut ne pas fonctionner pour certains clients e-mail comme AOL). Un autre exemple est la liste des réglages importants avec

```
netsh diag show all
```

XII-B-1 - L'interface graphique

Afin de simplifier les tâches de diagnostic, il est possible d'utiliser l'interface du Centre d'Aide et de support qui peut être appelée en tapant

```
netsh diag gui
```

Grâce à cette interface, vous pouvez effectuer de nombreux tests de diagnostic en une seule opération. La Figure 1 montre les fonctions disponibles.

Figure 1. Réglages pour l'interface graphique de Netsh diag



XII-B-2 - Le contexte « netsh interface ip »

Ce contexte contient des fonctions intéressantes pour tous les utilisateurs. Elle permet d'ajouter, supprimer, modifier et afficher les différentes adresses IP et les réglages TCP/IP. Le tableau II dresse la liste des différentes fonctions. Vous trouverez plus de détails et des fonctions supplémentaires dans cette [référence Microsoft](#). Vous pouvez par exemple utiliser netsh pour afficher les réglages TCP/IP avec la commande

```
netsh interface ip show config
```

Vous obtiendrez la même chose en entrant

```
netsh int ip show config
```

Tableau II. Quelques commandes pour « netsh interface ip »

Contexte	Sous-contexte	Fonction
interface ip ou int ip	set address	Configure une adresse IP et une passerelle par défaut sur une interface spécifiée.
	show address	Affiche l'adresse pour l'interface spécifiée.

XII-B-3 - Réinitialiser le protocole Internet (TCP/IP)

Un autre exemple de l'utilisation du contexte *Netsh Internet Ip* est la réinitialisation de TCP/IP. Ces réglages peuvent avoir été corrompus pour plusieurs raisons, notamment une infection de spyware. *Netsh* contient une commande qui va réinitialiser la pile TCP/IP aux réglages originaux qui existaient quand le système d'exploitation a été installé. Ces réglages ne sont peut-être pas les plus à jour pour votre système, mais ils vont vous permettre de tout reconfigurer après une infection de spyware ou tout autre problème. Pour réinitialiser TCP/IP, il faut utiliser la commande

```
netsh int ip reset {fichierjournal}
```

Remarquez que vous devez indiquer le nom d'un fichier dans lequel sera placé le journal des actions. Ainsi, si le fichier journal est *C:\tcplog.txt*, la commande sera

```
netsh int ip reset C:\tcplog.txt
```

Une description détaillée de la réinstallation de TCP/IP est donnée dans cette [référence Microsoft](#). Consultez également le paragraphe Winsock ci-dessous.

XII-C - Le contexte « netsh winsock »

Le Service Pack 2 pour Windows XP inclut quelques ajouts à la suite Netsh. Ils sont [décrits ici](#) et intègrent un nouvel outil pour réparer le socket réseau Windows ou « winsock ». Désinstaller un spyware ou une application mal programmée peut corrompre les réglages winsock et entraîner une perte de la connectivité réseau. Il existe deux commandes pour gérer les réglages. Pour afficher une liste des différents services installés (LSP, BSP, NSP), utilisez

```
netsh winsock show catalog
```

Cette liste n'est pas très compréhensible pour les utilisateurs lambda, mais elle peut être utile pour les utilisateurs plus expérimentés. Pour réinitialiser winsock à la configuration par défaut, utilisez

```
netsh winsock reset catalog
```

Notez bien que tous les programmes installés utilisant la connexion Internet peuvent être partiellement désactivés par cette action et devront être réinstallés. Cela peut être le cas des logiciels qui possèdent une fonctionnalité de mise à jour automatique et les programmes antivirus qui utilisent des serveurs proxy. Par conséquent, cette commande devrait être utilisée uniquement pour des cas où le degré de corruption la rend nécessaire. Consultez [cette référence](#) pour des méthodes alternatives.

XII-D - Netsh Firewall Helper dans Windows XP SP2 (assistance pare-feu)

À la sortie du Service Pack 2, Microsoft a modifié le fonctionnement du pare-feu dans Windows XP. Ils ont également ajouté à Netsh la possibilité d'effectuer des configurations avancées du pare-feu grâce au nouveau contexte « netsh firewall » que Microsoft appelle le « Firewall Helper » (l'assistance pare-feu). [Cet article de la base de connaissance](#) explique son utilisation pour résoudre les problèmes de pare-feu dans XP SP2. Microsoft indique qu'avec le Firewall Helper, vous pouvez maintenant :

- Configurer l'état standard du Pare-feu Windows. (Inactif, Activé et Activé sans exception.)
- Configurer les ports devant être ouverts.
- Configurer les ports pour permettre un accès global ou pour limiter l'accès au sous-réseau local.
- Définir les ports à ouvrir sur toutes les interfaces ou sur une interface spécifique.
- Configurer les options de connexion.
- Configurer les options de traitement du protocole ICMP.
- Ajouter ou supprimer des programmes dans la liste des exceptions.

La quantité de commandes possible est assez importante, mais les deux principaux sous-contextes sont

```
netsh firewall set
```

et

```
netsh firewall show
```

Une liste des commandes possibles est disponible dans [l'article de la base de connaissance](#) précédemment mentionné.

XIII - TCP/IP et outils Réseau



Ce chapitre est une traduction de la page <http://commandwindows.com/tcpiputil.htm> dont l'auteur original est Victor Laurie.

Windows XP possède un ensemble de commandes utiles pour configurer et tester les connexions Internet et LAN. Vous trouverez dans ce chapitre la description de quelques outils réseau qui peuvent être utiles pour tous les utilisateurs d'ordinateurs personnels.

Il existe des dizaines d'outils pour le réseau dans Windows XP (regardez par exemple cette [référence de l'invite de commande](#)). La plupart sont spécialisés et sont principalement intéressants pour les professionnels qui gèrent des grands réseaux. Beaucoup, cependant, sont en rapport avec Internet et certains d'entre eux peuvent être utiles pour les utilisateurs moyens. Les outils intéressants dont nous allons parler ici sont quelques utilitaires TCP/IP. (Allez [sur cette page](#) pour une liste complète). **TCP/IP** fait référence à un ensemble de protocoles utilisés pour les connexions Internet et dans beaucoup de réseaux. La description de TCP/IP n'a pas sa place dans ce chapitre, mais plus de détails sont disponibles sur [cette page](#). Heureusement, il n'est pas nécessaire de comprendre les détails gares de TCP/IP pour pouvoir utiliser les outils expliqués ici.

Tous les outils sont exécutés en ouvrant une invite de commande et en saisissant la commande appropriée. Allez dans **Démarrer-Exécuter** et tapez « cmd » pour ouvrir une invite de commande.

XIII-A - Outil de configuration IP de Windows (ipconfig)

L'outil de configuration IP (*ipconfig*) est l'équivalent en ligne de commande de l'accessoire « Winipcfg » qui existait dans Windows 9X/Me. Cet outil est utilisé pour afficher les valeurs de configuration réseau TCP/IP. Pour l'ouvrir, tapez

```
ipconfig
```

dans l'invite de commande. Si vous êtes directement connecté à Internet, vous allez ainsi obtenir votre adresse IP. (Pour une description de ce qu'est une IP, [rendez-vous ici](#)). La figure ci-dessous montre le résultat pour une connexion haut débit où l'IP est assignée automatiquement. Ici, l'IP est l'adresse temporaire de votre ordinateur sur l'Internet.

```

C:\WINDOWS\system32\cmd.exe
C:\>ipconfig

Configuration IP de Windows

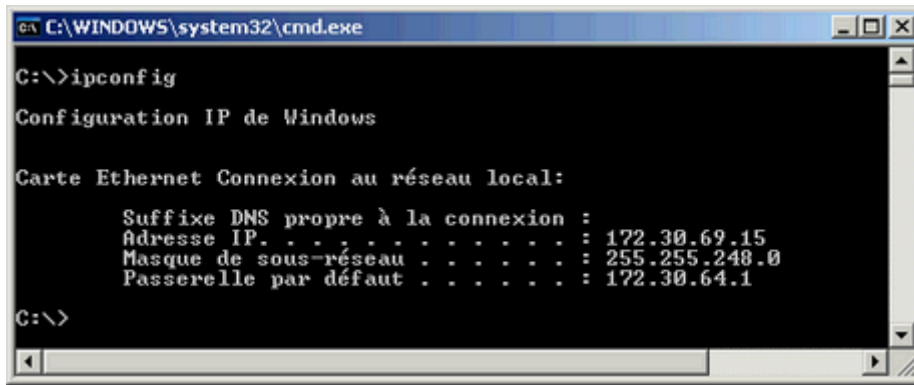
Carte Ethernet Connexion au réseau local:

    Suffixe DNS propre à la connexion :
    Adresse IP. . . . . : 172.30.69.15
    Masque de sous-réseau . . . . . : 255.255.248.0
    Passerelle par défaut . . . . . : 172.30.64.1

C:\>

```

Si vous êtes dans un réseau local et que vous utilisez un routeur, l'information sera différente. Vous n'obtiendrez pas l'IP correspondant à l'adresse réseau sur l'Internet. (Pour obtenir l'IP visible sur Internet, rendez-vous sur un site comme [DSL Reports Whois](#)). L'IP obtenu par ipconfig sera dans ce cas l'adresse *locale* sur le réseau. Cette information peut être utile pour diagnostiquer les problèmes réseau. La commande donne également la « passerelle » ou l'adresse du routeur sur un réseau local. L'image ci-dessous montre le résultat.



```
C:\WINDOWS\system32\cmd.exe
C:\>ipconfig

Configuration IP de Windows

Carte Ethernet Connexion au réseau local:
    Suffixe DNS propre à la connexion :
    Adresse IP. . . . . : 172.30.69.15
    Masque de sous-réseau . . . . . : 255.255.248.0
    Passerelle par défaut . . . . . : 172.30.64.1

C:\>
```

XIII-B - Options d'IPConfig

Il y a plusieurs commutateurs pour *ipconfig* qui ajoute des fonctions. Ces commutateurs sont utilisés en entrant

```
ipconfig /{commutateur}
```

Pour obtenir la liste des options possibles, entrez

```
ipconfig /?
```

ou

```
ipconfig -?
```

Les options sont visibles dans la figure ci-dessous. Les options les plus intéressantes pour une utilisation au quotidien sont « release » et « renew ». Remarquez que les adresses IP sont en général assignées ou « louées » pour une période définie, le plus souvent pour un jour ou plus. Il arrive parfois que les adresses IP ne soient plus valides ou soient en conflit. Ces problèmes peuvent souvent être résolus en commençant par libérer l'adresse IP (release) puis en la renouvelant (renew). Parfois les modems câble ou DSL qui semblent désactivés peuvent être restaurés de cette manière. Si vous voyagez et que vous utilisez des connexions haut débit dans des endroits différents, vous vous apercevrez que cette procédure est souvent nécessaire.

```
C:\>ipconfig /?

UTILISATION :
    ipconfig [/? | /all | /renew [carte] | /release [carte] |
        /flushdns | /displaydns | /registerdns |
        /showclassid carte |
        /setclassid carte [ID de classe] ]

où :
    carte          Nom de connexion
                   (caractères génériques * et ? autorisés, voir les exemples)

Options :
    /?             Affiche ce message d'aide.
    /all           Affiche toutes les informations de configuration.
    /release       Libère l'adresse IP pour la carte spécifiée.
    /renew         Renouvelle l'adresse IP pour la carte spécifiée.
    /flushdns      Vide le cache de la résolution DNS.
    /registerdns    Actualise tous les baux DHCP et réinscrit les noms DNS.
    /displaydns    Affiche le contenu du cache de la résolution DNS.
    /showclassid   Affiche tous les ID de classe DHCP autorisés pour la carte.
    /setclassid    Modifie l'ID de classe DHCP.
```

Pour obtenir la liste détaillée des paramètres, vous pouvez utiliser la commande

```
ipconfig /all
```

À moins que vous ne soyez expert en réseau, ceci devrait être suffisant.

Les options *flushdns* et *displaydns* sont parfois utiles au quotidien et elles sont décrites sur [cet autre site](#).

XIII-C - Ping

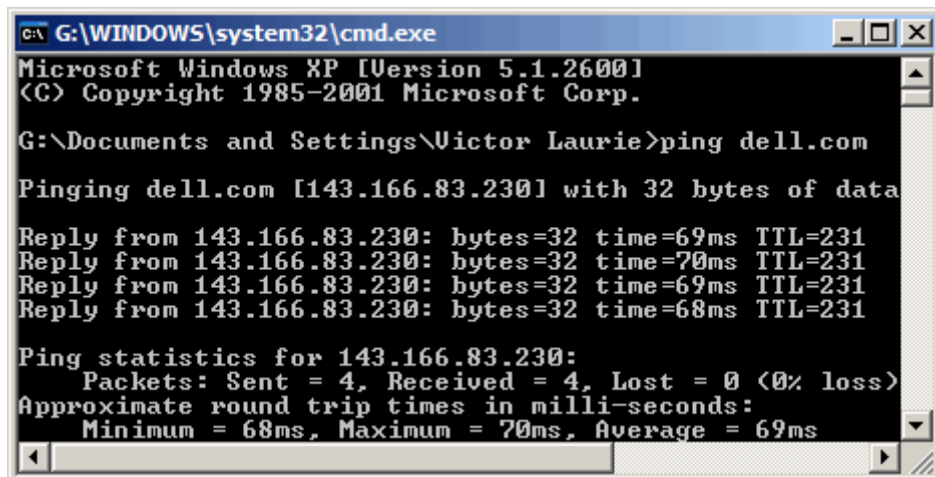
Ping est un vieux outil Unix qui existe depuis bien longtemps, mais beaucoup d'utilisateurs ne sont pas familiers avec la version Windows. *Ping* envoie un paquet à un hôte Internet spécifié ou à un ordinateur du réseau et mesure le temps de réponse. L'ordinateur cible renverra (avec un peu de chance) un signal. Cela permet de déterminer la qualité de la connexion vers un autre site. Vous obtiendrez également une adresse IP qui correspond à l'URL « conviviale » (consultez [cette page](#) pour plus d'informations sur les IP et les URL). Pour utiliser ping, ouvrez une invite de commande (ou DOS dans Windows 9X/Me) et tapez :

```
ping nom_de_l_hote
```

Par exemple, pour pinguer le site de Dell, entrez :

```
ping dell.com
```

Remarquez qu'il faut utiliser le nom de l'hôte et non l'URL complète. Le préfixe « http:// » n'est jamais utilisé. Pour pinguer la plupart des sites, il n'y a pas non plus besoin de « www ». Par convention, des paquets de 32 octets vont être envoyés 4 fois. Vous allez obtenir un résultat comme ceci :



```
C:\G:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

G:\Documents and Settings\Victor Laurie>ping dell.com

Pinging dell.com [143.166.83.230] with 32 bytes of data:

Reply from 143.166.83.230: bytes=32 time=69ms TTL=231
Reply from 143.166.83.230: bytes=32 time=70ms TTL=231
Reply from 143.166.83.230: bytes=32 time=69ms TTL=231
Reply from 143.166.83.230: bytes=32 time=68ms TTL=231

Ping statistics for 143.166.83.230:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 68ms, Maximum = 70ms, Average = 69ms
```

Ce résultat m'indique que l'IP de *dell.com* est 143.166.83.230 (pour le moment, en tout cas. Cette adresse peut changer). Cela m'indique aussi que le temps moyen de trajet des paquets a été de 69 millisecondes, ce qui signifie que je dispose d'une bonne connexion à *dell.com*. Les temps de réponse longs de plusieurs centaines de millisecondes indiquent une connexion lente. Remarquez que certains sites majeurs comme *microsoft.com* n'aiment pas être pingués et bloquent les pings. Dans ce cas, vous obtiendrez un message « Délai d'attente de la demande dépassé ».

En plus d'être utilisé sur Internet, Ping est souvent utilisé pour tester les connexions sur les réseaux locaux. Vous trouverez plus de détails à ce sujet dans cet [article Microsoft](#).

XIII-D - Tracert

Tracert (traceroute) est un autre vieux outil emprunté à Unix. Le chemin entre deux ordinateurs sur Internet n'est pas en ligne droite, mais consiste en de nombreux segments ou « hops » d'un ordinateur intermédiaire à un autre. *Tracert* affiche chaque étape du chemin emprunté. Il peut être intéressant de voir jusqu'à quel point il est compliqué. Le temps pour chaque « hop » et l'adresse IP de chaque ordinateur intermédiaire sont affichés. *Tracert* affiche jusqu'à

30 « hops ». Cela est particulièrement utile pour savoir si un segment particulier provoque une mauvaise connexion ou une lenteur. La commande peut s'écrire par exemple « `tracert dell.com` ».

XIII-E - Pathping

Cette commande combine les fonctions de *Ping* et de *Tracert*. *Pathping* va tout d'abord lister le nombre de « hops » nécessaires pour atteindre l'adresse que vous testez puis va envoyer plusieurs pings à chaque routeur entre vous et la destination. Après cela, la commande calcule les résultats basés sur les paquets renvoyés par chaque routeur. Comme pathping affiche la proportion de paquets perdus pour chaque routeur ou lien, vous pouvez déterminer quels routeurs ou sous-réseaux ont des problèmes de réseau. Notez que le processus complet peut prendre entre 5 et 10 minutes parce que beaucoup de pings sont envoyés. Cette commande dispose de plusieurs options pour modifier le processus. Vous pouvez voir ces options en tapant « `pathping /?` » dans l'invite de commande.

XIII-F - Netstat

Netstat affiche les connexions TCP actives et les ports sur lesquels l'ordinateur est en écoute, les statistiques Ethernet, la table de routage IP ainsi que les statistiques pour les protocoles IP, ICMP, TCP et UDP. Elle dispose de nombreuses options pour afficher plusieurs propriétés du réseau et des connexions TCP. (*Un point à noter : les options sont préfixées par un tiret et non pas un slash*). Vous trouverez plus d'informations [sur cette page](#). Vous pouvez par exemple utiliser *Netstat* pour déterminer si un spyware ou un ver a établi des connexions sans que vous ne le sachiez. La commande « `netstat -a` » affiche toutes les connexions. La commande « `netstat -b` » affiche les fichiers exécutables qui ont ouvert des connexions. Vous pouvez voir toutes les options et la syntaxe dans l'image ci-dessous.

```
Affiche les statistiques de protocole et des connexions réseau TCP/IP actuelles.
NETSTAT [-a] [-b] [-e] [-n] [-o] [-p protocole] [-r] [-s] [-v] [intervalle]

-a      Affiche toutes les connexions et les ports d'écoute.
-b      Affiche les fichiers exécutables impliqués dans la création de chaque connexion ou
        port d'écoute. Certains fichiers exécutables connus peuvent héberger
        plusieurs composants indépendants. Dans ce cas, la
        séquence des composants impliqués dans la création de la connexion
        ou du port d'écoute est affichée, et le nom du fichier exécutable
        apparaît entre crochets [] en bas, le nom du composant appelé apparaît en haut,
        et ainsi de suite jusqu'à ce que TCP/IP soit atteint. Remarquez que cette option
        peut prendre du temps et échouera si vous ne disposez pas des autorisations
        suffisantes.
-e      Affiche les statistiques Ethernet. Cette option peut être combinée avec l'option
        -s.
-n      Affiche les adresses et les numéros de port au format numérique.
-o      Affiche l'identificateur du processus propriétaire associé à chaque connexion.
-p protocole Affiche les connexions pour le protocole spécifié ; protocole
        peut être une des valeurs suivantes : TCP, UDP, TCPv6 ou UDPv6. S'il est utilisé av
        ec l'option
        -s pour afficher les statistiques par protocole, le protocole peut être une des vale
        urs suivantes :
        IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP ou UDPv6.
-r      Affiche la table de routage.
-s      Affiche les statistiques par protocole. Par défaut, les statistiques sont
        affichées pour IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP et UDPv6.
        L'option -p peut être utilisée pour spécifier un sous-jeu de la valeur par défaut.
        Lorsqu'elle est utilisée avec -b, cette option affichera la séquence des
        composants impliqués dans la création de la connexion ou du port d'écoute
        pour tous les fichiers exécutables.
        intervalle Affiche régulièrement les statistiques sélectionnées, en faisant une pause pendant l
        e nombre de secondes spécifié par l'intervalle
        entre chaque affichage. Appuyez sur CTRL+C pour arrêter l'affichage
        des statistiques. Si l'intervalle est omis, netstat n'affichera les
        informations de configuration actuelle qu'une seule fois.
```

XIII-G - Nslookup

Cette commande aide à diagnostiquer les problèmes sur l'infrastructure DNS (Domain Name System) et dispose de nombreuses sous-commandes. Elles s'adressent principalement aux administrateurs systèmes. Le principal intérêt pour les utilisateurs standard est de trouver le nom d'un ordinateur à partir de son adresse IP. Par exemple si vous voulez savoir qui est « 216.109.112.135 », entrez « `nslookup 216.109.112.135` » et vous devriez trouver qu'il s'agit (ou s'agissait du moins) d'un ordinateur de chez Yahoo. Mon pare-feu conserve la trace de toutes les IP qui ont tenté de tester mon ordinateur et je regarde parfois quelques-uns pour savoir qui ils sont. (Il existe également des sites de recherche *WhoIs* sur le Web, comme cela a déjà été indiqué dans la section *Ipconfig*)

XIII-H - Netsh

La console des services réseau est composée d'un ensemble d'outils. Ils sont décrits plus précisément sur [une autre page](#).

XIV - Utilitaire de système de fichiers - Fsutil.exe



Ce chapitre est une traduction de la page <http://commandwindows.com/fsutil.htm> dont l'auteur original est Victor Laurie.

L'utilitaire de système de fichiers « fsutil » est une suite d'outils en ligne de commande pour afficher et gérer certaines propriétés des lecteurs et des fichiers. Quelques applications sont décrites ici.

Fsutil est un outil avancé qui s'adresse tout d'abord aux administrateurs, mais des utilisateurs expérimentés trouveront également qu'il possède un certain nombre d'applications possibles. Je vais décrire ici quelques-unes de ces applications qui me semblent intéressantes. Cet outil est présent dans Windows XP et Vista et est essentiellement utilisé sur les systèmes NTFS. Pour l'utiliser, vous devez posséder les privilèges d'administrateur.

XIV-A - Sous-commandes

Fsutil contient une suite de sous-commandes qui sont listées dans le tableau I. Chacune de ces sous-commandes peut avoir elle-même des sous-commandes. Beaucoup d'entre elles sont assez spécialisées, mais les plus intéressantes sont décrites dans les paragraphes suivants. Windows Vista possède deux sous-commandes additionnelles qui ne sont pas présentes dans XP. Elles sont indiquées dans le tableau. Elles sont indiquées par souci d'exhaustivité, mais présentent un intérêt limité pour la plupart des utilisateurs.

Tableau I. Sous-commandes de Fsutil

Sous-commande	Description
behavior	Gère l'option de génération des noms au format 8.3 et l'option de mise à jour de la date du dernier accès. Gère la quantité d'espace disque réservé à la MFT (Master File Table).
dirty	Interroge ou définit le bit d'impureté
file	Permet de rechercher un fichier par son identificateur de sécurité, interroge les plages allouées d'un fichier, définit le nom court du fichier, la longueur des données valides d'un fichier ou les zones égales à zéro d'un fichier.
fsinfo	Affiche la liste des lecteurs, donne des informations sur le type du lecteur, des informations sur le volume, interroge les informations spécifiques aux volumes NTFS, ou

	donne les statistiques du système de fichier.
hardlink	Crée un lien réel (hard link).
objectid	Gère les identificateurs d'objet.
quota	Gère les quotas de disque sur les volumes NTFS
repair (Vista)	Gestion de réparation spontanée.
reparsepoint	Interroge ou supprime des points d'analyse.
sparse	Gestion des fichiers fragmentés.
transaction (Vista)	Gestion des transactions.
usn	Gestion du journal USN (Update Sequence Number).
volume	Démonte un volume ou l'interroge pour connaître la quantité d'espace libre disponible sur un disque.

XIV-B - Exemples d'applications de Fsutil

Vous trouverez ici une sélection des utilisations de *Fsutil* que j'ai souvent vues mentionnées sur le Web.

XIV-B-1 - Utilisations de la sous-commande « behavior »

La sous-commande « behavior » est peut être l'une de celles qu'un utilisateur ordinaire va le plus utiliser. Il existe plusieurs fonctionnalités dans le système NTFS qui peuvent être configurées pour accélérer les systèmes. Et des astuces en rapport avec cette commande sont souvent mentionnées sur le Web ou dans les livres. La sous-commande « behavior » possède deux sous-commandes. Il s'agit de « query », qui affiche les valeurs actuelles des réglages et « set », qui est utilisée pour changer les réglages. Le tableau II indique les options disponibles. Les options qui sont disponibles seulement dans Vista sont indiquées. Les commandes sont de la forme

```
fsutil behavior query {option} {valeur}
```

et

```
fsutil behavior set {option} {valeur}
```

Les commandes utilisant « set » peuvent modifier des éléments du registre et peuvent nécessiter un redémarrage pour être prises en compte.

Tableau II. Options de « fsutil behavior »

Option	Valeurs possibles
allowextchar	0 ou 1
disable8dot3	0 ou 1
disablecompression (Vista)	0 ou 1
disablelastaccess	0 ou 1
disableencryption (Vista)	0 ou 1
encryptpagingfile (Vista)	0 ou 1
mftzone	0 à 4
memoryusage (Vista)	0, 1, 2
quotanotify	1 à 4294967295 secondes
SymlinkEvaluation (Vista)	Valeurs diverses

XIV-B-1-a - Consultez les réglages actuels


Pour afficher les réglages actuels sur un système, utilisez une commande avec « query » comme :

```
fsutil behavior query disablelastaccess
```

XIV-B-1-b - Désactiver les noms de fichiers courts pour accélérer Windows

Une astuce qui est souvent mentionnée est la désactivation de la création des noms de fichiers au format court 8.3. Par défaut, Windows XP (NTFS) et Vista créent des noms de fichiers à l'ancien format 8.3 en plus du nom réel du fichier. Ceci permet d'assurer la compatibilité avec certains programmes ont besoin de l'ancien format de noms datant de DOS. Si vous êtes certain que vous n'avez aucun programme 16-bits ou des programmes qui nécessitent l'ancien format, la désactivation de cette option peut accélérer votre système. La commande est :

```
fsutil behavior set disable8dot3 1
```

Remarquez que les  **variables d'environnement** de Windows XP %TEMP% et %TMP% utilisent les noms courts. Les installations de programmes utilisent parfois ces variables. Si ce changement entraine des problèmes et que vous souhaitez réactiver la création de noms courts, utilisez la commande :

```
fsutil behavior set disable8dot3 0
```

XIV-B-1-c - Désactivation de l'heure de dernière modification pour accélérer Windows

On rencontre souvent une autre recommandation : il s'agit de désactiver l'option qui garde la trace du dernier accès à un fichier. Si vous évitez au système de lire et écrire cette information, vous pourrez accélérer l'explorateur Windows. La commande à utiliser est :

```
fsutil behavior set disablelastaccess 1
```

Notez bien que certains programmes de sauvegarde peuvent avoir besoin de cette information. Si vous souhaitez restaurer cette option, la commande est :

```
fsutil behavior set disablelastaccess 0
```

XIV-B-1-d - Changer la taille de la MFT (Master File Table)

NTFS utilise une table de fichiers principale (Master File Table : MFT) pour stocker des informations concernant les dossiers et fichiers. Même les petits fichiers y sont inclus. Par défaut, 12.5 % du volume est alloué à la MFT. Cette allocation par défaut devrait suffire dans la plupart des cas, sauf si vous avez un petit volume ou un très grand nombre de fichiers. Si vous avez besoin de plus de place pour la MFT, vous pouvez utiliser la commande

```
fsutil behavior set mftzone n
```

Le paramètre « n » est compris entre 0 et 4. Le tableau III indique l'allocation correspondant à chaque valeur de « n ».

Tableau III. Allocations d'espace MFT

valeurs de n	Espace alloué pour la MFT
0	Aucun réglage. Utilise la valeur par défaut 12.5 %
1	12.5 %
2	25 %
3	37.5 %
4	50 %

XIV-B-2 - Afficher quelques propriétés du lecteur

Fsutil possède des méthodes très poussées pour gérer les disques durs, mais je vais n'en mentionner que quelques-unes parmi les plus intéressantes.

XIV-B-2-a - Obtenir la liste des lecteurs d'un ordinateur

Il existe d'autres manières de faire, mais vous pouvez obtenir une liste des lettres de lecteurs en utilisant la commande :

```
fsutil fsinfo drives
```

XIV-B-2-b - Obtenir le type de lecteur

Vous pouvez obtenir le type d'un lecteur donné avec la commande :

```
fsutil fsinfo drivetype D:
```

La commande renvoie « Lecteur fixe », « Lecteur de CD-ROM » ou « Lecteur amovible ».

XIV-B-2-c - Obtenir des informations générales sur un volume

Vous pouvez obtenir une liste de plusieurs paramètres de volume avec :

```
fsutil fsinfo volumeinfo C:
```

XIV-B-2-d - Obtenir la quantité d'espace libre sur un lecteur

Pour déterminer la quantité d'espace libre sur un lecteur, utilisez la commande

```
fsutil volume diskfree C:
```

XV - La console de récupération de Windows XP



Ce chapitre est une traduction de la page <http://commandwindows.com/recovery.htm> dont l'auteur original est Victor Laurie.

La console de récupération Windows XP est un exemple d'accessoire en ligne de commande. Elle est d'une aide précieuse si Windows a besoin d'une réparation et ne démarre pas, en particulier pour les machines NTFS.

Même si vous n'avez pas l'intention d'utiliser la ligne de commande, il est utile de se familiariser avec la Console de récupération. Dans le cas où un problème empêche Windows XP de démarrer, cet utilitaire peut vous permettre de réparer un système corrompu ou de récupérer vos précieux fichiers personnels. Sur les systèmes FAT, une **disquette de démarrage DOS** peut être utilisée pour accéder à un système rencontrant des problèmes. Mais les systèmes NTFS, qui sont courants aujourd'hui ne sont pas accessibles depuis DOS. Comme indiqué sur [un autre site](#), il est possible de créer un CD bootable fournissant des services d'urgences. Ces CD utilisent Linux ou des lecteurs NTFS spéciaux. Cependant, Windows XP est fourni avec ces propres accessoires pour faire cela. La console de récupération fonctionne à la fois sur les disques FAT et NTFS, mais est particulièrement utile pour les systèmes utilisant NTFS. Il semble que cet outil soit méconnu, car Microsoft semble vouloir le garder obscur et difficile à utiliser. Dans ce chapitre, je vais essayer de clarifier son utilisation et d'expliquer comment cette console peut aider tous les propriétaires d'ordinateurs.

XV-A - Accéder à la console de récupération avec un CD de Windows.

Si vous avez un disque d'installation de Windows, vous pouvez accéder à la console en démarrant sur le CD de Windows. Après un certain temps de chargement, vous devriez obtenir un écran « Installation de Windows XP - Bienvenue ! ». Appuyez sur la touche « r » et une invite va apparaître en vous demandant le mot de passe administrateur. Les utilisateurs de la version familiale ou ceux qui n'ont pas de mot de passe administrateur peuvent ne rien saisir à cette étape. (Remarquez que ce mot de passe n'est pas forcément le même que celui d'un compte utilisateur, même si celui-ci possède les privilèges d'administrateur). L'écran va ensuite afficher une liste numérotée de toutes vos installations de Windows (la plupart des gens n'en ont qu'une). La console va vous demander « Sur quelle installation de Windows voulez-vous ouvrir une session ». Si vous n'avez pas un système multiboot, entrez le chiffre « 1 ». Vous devez entrer un chiffre. NE tapez PAS directement Entrée. Après un temps de chargement, la console de récupération va se charger et vous fournir une invite de commande où plusieurs tâches système peuvent être effectuées.

XV-B - Accéder à la console de récupération sans un CD de Windows.

Souvent, les vendeurs d'ordinateurs ne fournissent pas le CD de Windows avec le système. Ils fournissent à la place un **CD de restauration** ou simplement une partition cachée. Ils *peuvent éventuellement* fournir le moyen d'accéder à la console de récupération, mais chaque propriétaire d'ordinateur doit vérifier sur sa configuration. Si, comme souvent, la console de récupération est absente, il existe un autre moyen de l'obtenir.

Microsoft fournit un logiciel pour créer un lot de six disquettes pour réinstaller Windows XP pour ceux qui n'ont pas le CD bootable. Les **détails sont ici**. Cela semble très peu connu, mais ces disquettes contiennent la console de récupération. Ainsi, créer cet ensemble de disquettes va permettre à ceux qui ne disposent pas du CD (mais qui ont un lecteur de disquette) la possibilité d'utiliser la console de récupération. C'est assez pénible ; vous devez attendre que le système lance le processus d'installation et charge chacune des six disquettes, mais s'il sauve votre système ça vaudra le coup. Une fois que vous êtes sur l'écran d'accueil, la procédure est la même que ci-dessus avec le CD.

Si vous n'avez ni le CD de Windows XP ni un lecteur de disquette, il existe des méthodes pour mettre des images de disquettes sur un CD bootable. Une méthode a été fournie par Dean Adams pour créer un CD bootable contenant la console de récupération. Vous pouvez la consulter sur [cette page](#).

XV-C - Commandes dans la console de récupération

Une fois dans la console de récupération, que peut-on faire avec ? Le tableau I liste les commandes disponibles. Beaucoup de ces commandes disposent d'options. Des détails sur la syntaxe et les fonctions sont disponibles sur [une autre page](#) ou peuvent être obtenus en entrant

```
{commande} /?
```

dans la console de récupération. Vous trouverez également des détails sur [cette page Microsoft](#). Remarquez que la console possède son propre interpréteur de commandes et que les commandes peuvent être différentes de celles portant le même nom dans une invite de commande traditionnelle. De plus, dans la configuration par défaut, certaines commandes sont désactivées ou ont des fonctionnalités limitées. La méthode pour enlever quelques-unes de ces restrictions est décrite dans le paragraphe suivant.

Tableau I. Commandes disponibles dans la console de récupération.

Commande	Description
Attrib	Change les attributs d'un fichier ou d'un dossier.
Batch	exécute les commandes spécifiées dans le fichier texte.
Bootcfg	Configuration et récupération du fichier de démarrage (boot.ini)
ChDir (Cd)	Affiche le nom du répertoire courant ou change le répertoire courant
Chkdsk	Vérifie un disque et affiche un rapport de l'état du disque
Cls	Vide l'écran
Copy	Copie un fichier dans un autre emplacement
Delete (Del)	Supprime un ou plusieurs fichiers
Dir	Affiche une liste des fichiers et sous-répertoires dans un répertoire.
Disable	Désactive un service système ou un pilote de périphérique.
Diskpart	Gère les partitions sur vos disques durs.
Enable	Démarre ou active un service système ou un pilote de périphérique.
Exit	Quitte la console de récupération et redémarre votre ordinateur
Expand	Extrait un fichier d'un fichier compressé.
Fixboot	Écrit un nouveau secteur de démarrage sur la partition spécifiée.
Fixmbr	Répare le Master Boot Record (MBR) sur le disque spécifié.
Format	Formate un disque.
Help	Affiche une liste des commandes que vous pouvez utiliser dans la console de récupération.
Listsvc	Liste les services et les pilotes disponibles sur l'ordinateur
Logon	Se connecte à une installation Windows
Map	Affiche l'affectation des lettres de lecteur.
Mkdir (Md)	Crée un répertoire.
More	Affiche un fichier texte.
Rename (Ren)	Renomme un fichier
Rmdir (Rd)	Supprime un répertoire
Set	Affiche et définit les variables d'environnement
Systemroot	Définit le répertoire comme le répertoire racine du système sur lequel vous êtes actuellement connecté.
Type	Affiche un fichier texte.

XV-D - Enlever les restrictions de la console.

La configuration par défaut de la commande contient un certain nombre de limitations. Je suppose que le but est de limiter les dégâts que les non-initiés peuvent causer. Cependant, cela réduit considérablement l'utilité de la console.

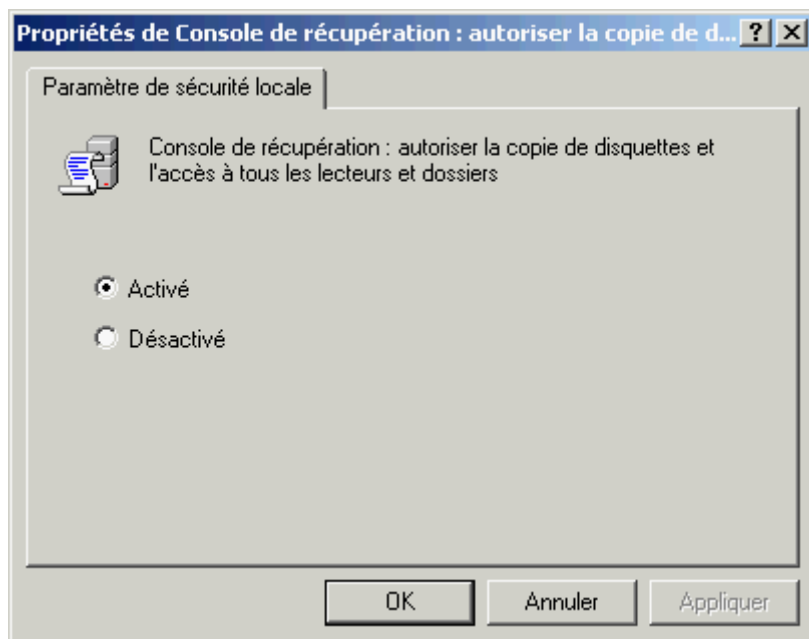
Ces restrictions sont par exemple l'interdiction d'accès à tous les répertoires sauf quelques-uns, l'impossibilité d'écrire sur une disquette et l'impossibilité d'utiliser des jokers dans les commandes. Ces restrictions peuvent être enlevées avec plusieurs méthodes. Elles sont décrites dans les trois prochains paragraphes. Remarquez que ces opérations doivent être faites depuis une connexion classique à Windows avec les privilèges administrateur. Il n'est pas possible de le faire directement depuis la console de récupération.

XV-D-1 - Utilisation de la console de stratégie de sécurité locale ou de stratégie de groupe

Cette méthode est accessible aux utilisateurs de Windows XP Professionnel et concerne une section de l'éditeur de stratégies de groupe appelé « Options de sécurité ». Une description est disponible dans cet [article Microsoft](#), mais ce n'est pas un modèle de clarté. (Les instructions Microsoft à ce sujet sont souvent confuses.) Il existe deux consoles de gestion qui peuvent être utilisées, l'éditeur de stratégie de groupe *gpedit.msc* ou sa sous-section *secpol.msc* (consultez ce site pour une [description des consoles de gestion](#)). Le moyen le plus simple d'ouvrir les réglages de sécurité locale est de taper

```
secpol.msc
```

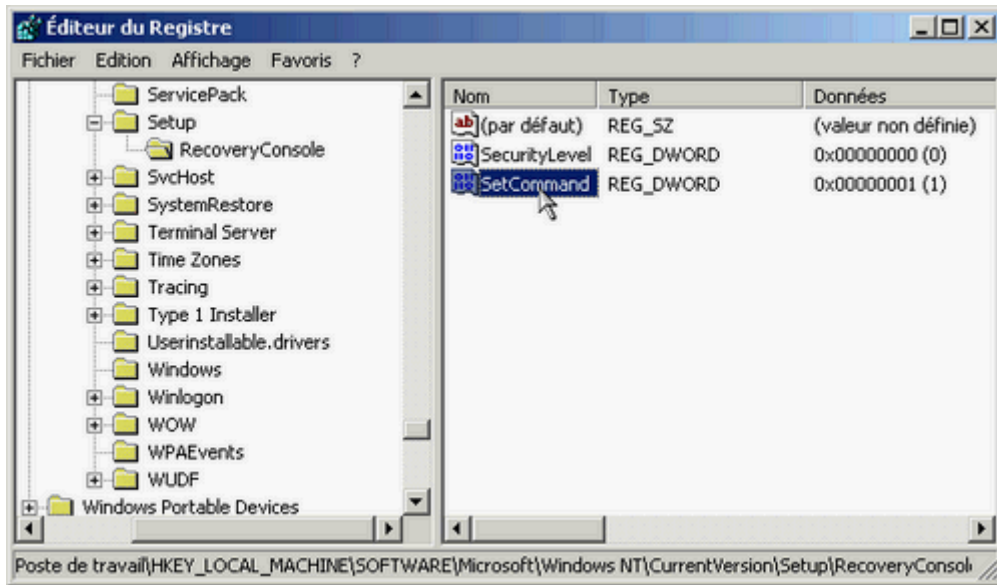
dans **Démarrer-Exécuter**. Cliquez ensuite sur « Stratégies locales-Options de sécurité » dans le panneau de gauche. Dans la liste qui apparaît dans le panneau de droite, trouvez la ligne « Console de récupération : autoriser la copie de disquettes et l'accès à tous les lecteurs et dossiers » et double-cliquez dessus. La boîte de dialogue ci-dessus va alors s'ouvrir. Cliquez sur le bouton « Activé » puis sur « OK ».



Le nom de cette option est trompeur. Il inclut bien plus que l'activation de l'écriture sur une disquette ou l'accès à tous les dossiers. Cette option, en réalité, vous donne la permission d'utiliser la commande « set » dans la console de récupération. Cette commande est utilisée pour enlever les restrictions et est décrite plus loin.

XV-D-2 - Modifier le Registre

La procédure indiquée ci-dessous fonctionne en modifiant une valeur dans le Registre. Si vous êtes familier avec l'édition du registre, la modification directe du registre est un autre moyen qui présente l'avantage d'être accessible aux possesseurs de la version familiale de Windows XP. Ouvrez *regedit* et trouvez la clé **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Setup\RecoveryConsole**. Dans le panneau de droite, cette clé devrait avoir une valeur « SetCommand ». Mettez sa valeur à 1. Ceci est illustré sur la figure ci-dessous. *Sauvegardez le registre avant d'effectuer le moindre changement.*



XV-D-3 - Un script pour effectuer la modification du Registre

Éditer le Registre en toute sécurité n'est pas à la portée de tout le monde. Pour ceux qui ne sont pas confiants quant à la modification du registre, j'ai créé un petit script qui effectue la modification du registre pour vous. Pour ce script, j'ai choisi le format INF. Vous devez probablement connaître les fichiers INF puisqu'il s'agit du moyen standard pour installer des drivers (pilotes). [Cliquez ici pour télécharger setcommand.inf](#) (zippé). Avant de l'utiliser, sauvegardez le registre ou créez un point de restauration. Dézippez le fichier là où vous le souhaitez puis effectuez un clic droit sur celui-ci pour ouvrir le menu contextuel. Cliquez sur « Installer » dans le menu. Et c'est tout. Le registre a été modifié automatiquement. N'utilisez pas ce fichier sur les systèmes Windows 98/Me.

XV-E - Utilisation de la commande « Set »

La modification du registre décrite précédemment ne rajoute pas réellement de nouvelles fonctionnalités puisqu'il faut encore effectuer une étape directement dans la console de récupération. Les modifications du registre ne changent rien. Elles vous donnent la *permission* d'effectuer des changements avec la commande « set ». Ceux qui sont habitués à l'invite de commande traditionnelle connaissent la commande « set » et les variables d'environnement. (Vous pouvez en apprendre plus à ce sujet en [lisant ceci](#)). Les possibilités de la console de récupération peuvent être améliorées par 4 commandes « set ». La première autorise l'accès à tous les fichiers et dossiers sur le système principal. Dans l'invite de la console de récupération, tapez

```
set AllowAllPaths = true
```

La seconde commande vous autorise à copier des fichiers vers des médias amovibles comme les disquettes. Tapez

```
set AllowRemovableMedia = true
```

La commande suivante vous permet d'utiliser des jokers dans les commandes comme « dir » et « del ». Tapez

```
set AllowWildCards = true
```

Enfin, la dernière commande vous permet de copier des fichiers sans confirmation d'écrasement si le fichier existe déjà. Tapez

```
set NoCopyPrompt = true
```

Vous devez mettre une espace avant et après le signe « égal » dans toutes les commandes « set » ou elles ne fonctionneront pas. Ces commandes ne tiennent pas compte de la casse.

XV-F - La seule commande à apprendre

Si vous ne voulez pas apprendre d'autres commandes pour utiliser la console de récupération, rappelez-vous de celle-ci :

```
chkdsk C: /r
```

Il y a beaucoup de témoignages sur le Web qui indiquent que cette commande a dépanné beaucoup de systèmes qui ne démarraient pas. Si vous utilisez la commande à partir des disquettes d'installation, *chkdsk* va signaler qu'il ne peut pas trouver le fichier *autochk.exe*. Lorsque la commande vous demande l'emplacement du fichier, indiquez **\\windows\\system32**. Cette commande va examiner complètement votre disque dur et peut durer assez longtemps donc laissez-lui du temps.

XVI - Commandes de la console de récupération



Ce chapitre est une traduction de la page  <http://commandwindows.com/recovery-console-commands.htm> dont l'auteur original est Victor Laurie.

L'invite de commande qui est disponible dans la console de récupération est différente de l'invite de commande dans le système Windows XP normal. Les commandes disponibles sont décrites dans ce chapitre.

La console de récupération ainsi que ces commandes **ont été expliquées dans un chapitre précédent**. Ce chapitre va vous donner des détails complémentaires sur les commandes.

Attrib

Change les attributs d'un fichier ou d'un dossier. Possède des paramètres différents de l'invite de commande. La syntaxe est :

```
attrib [+r|-r] [+s|-s] [+h|-h] [+c|-c] [[lecteur:]][chemin] nom-de-fichier]
```

Les commutateurs sont décrits dans le tableau ci-dessous.

Commutateur	Fonction
r	Contrôle l'attribut « Lecture seule ».
s	Contrôle l'attribut « Système ».
h	Contrôle l'attribut « Caché ».
c	Contrôle l'attribut de fichier compressé.

Batch

Exécute les commandes dans le fichier texte spécifié. Si vous le souhaitez, les résultats peuvent être envoyés dans un fichier de sortie. Non disponible en-dehors de la console de récupération. La syntaxe est :

```
batch fichier_entree.txt [fichier_sortie]
```

Bootcfg

Utilisé pour configurer le fichier de démarrage *boot.ini*. Possède des paramètres différents de l'invite de commande. La syntaxe est :

```
bootcfg /parametre
```

Les paramètres possibles sont donnés dans le tableau ci-dessous.

Paramètre	Fonction
add	Ajoute une installation de Windows dans la liste de démarrage
copy	Sauvegarde <i>boot.ini</i>
default	Choisit l'entrée de démarrage par défaut
disableredirect	Désactive la redirection dans le chargeur de démarrage
list	Affiche la liste des éléments présents dans la liste de démarrage
rebuild	Parcourt toutes les installations de Windows
redirect	Active la redirection dans le chargeur de démarrage
scan	Recherche les installations de Windows sur tous les disques

CD (Chdir)

Affiche le nom du répertoire courant ou change le répertoire courant. Aucun commutateur. Se comporte légèrement différemment de l'invite de commande.

Chkdsk

Vérifie un disque et affiche un rapport de l'état du disque. Possède des paramètres différents de l'invite de commande. La syntaxe est :

```
chkdsk [lecteur:] [/p] [/r]
```

L'option /p effectue une vérification exhaustive d'un lecteur sans apporter de changement. Le commutateur /r effectue la même vérification, mais récupère également les informations lisibles. (Notez qu'il existe des sites Internet erronés qui suggèrent d'utiliser des options qui sont uniquement disponibles dans l'invite de commande standard.)

Cls

Vide l'écran. A la même fonction que dans l'invite de commande.

Copy

Copie un fichier dans un autre emplacement. Ne possède pas les options présentes dans l'invite de commande.

Del

Supprime un fichier. Ne possède pas les options présentes dans l'invite de commande.

Dir

Affiche une liste des fichiers et sous-répertoires dans un répertoire. Ne possède pas les options présentes dans l'invite de commande.

Disable

Désactive un service système ou un pilote de périphérique. Cette commande est accessible uniquement via la console de récupération. La syntaxe est :

```
disable {[nom_du_service] | [nom_du_pilote_de_peripherique]}
```

Diskpart

Crée ou supprime des partitions sur un disque dur. Cette commande est différente de la version très restreinte disponible dans l'invite de commande normale. La syntaxe est :

```
diskpart [/add | /delete] [nom_du_peripherique | nom_du_lecteur | nom_de_la_partition] [taille]
```

Enable

Démarre ou active un service système ou un pilote de périphérique. Cette commande est accessible uniquement via la console de récupération. La syntaxe est :

```
enable {service_name | device_driver_name} [startup_type]
```

Exit

Quitte la console de récupération et redémarre votre ordinateur

Expand

Extrait un fichier d'un fichier compressé. Possède des paramètres différents de l'invite de commande. La syntaxe est :

```
expand source [/F:nom_de_fichier] [destination] [/d] [/y]
```

Le commutateur /F:nom_du_fichier vous permet d'extraire un seul fichier nommé « nom_du_fichier ». Pour afficher tous les fichiers contenus dans un fichier cabinet (.cab) sans les extraire, utilisez le commutateur /d. Le commutateur /y supprime la confirmation d'écrasement d'un fichier par un fichier extrait.

Fixboot

Écrit un nouveau secteur de démarrage sur la partition système. Cette commande est accessible uniquement via la console de récupération. La syntaxe est :

```
fixboot [lecteur]
```

Fixmbr

Répare le Master Boot Record (MBR) sur le disque spécifié. Cette commande est accessible uniquement via la console de récupération. La syntaxe est :

```
fixmbr [nom_de_peripherique]
```

Si vous ne spécifiez pas de *nom_de_périphérique*, un nouveau Master Boot Record sera écrit sur le périphérique de démarrage, il s'agit du lecteur sur lequel votre système primaire est chargé.

Format

Formate le lecteur spécifié avec le système de fichier spécifié. La syntaxe est :

```
format [lecteur:] [/q] [/fs:systeme-de-fichier]
```

Le commutateur /q permet le formatage rapide. Le commutateur /fs: permet de choisir le système de fichier.

Help

Affiche une liste des commandes que vous pouvez utiliser dans la console de récupération.

Listsvc

Liste les services et les pilotes disponibles sur l'ordinateur. Cette commande est accessible uniquement via la console de récupération.

Logon

Se connecte à une installation Windows. Cette commande est accessible uniquement via la console de récupération.

Map

Affiche l'affectation des lettres de lecteur. Cette commande est accessible uniquement via la console de récupération.

Md (mkdir)

Crée un répertoire ou un sous-répertoire.

More

Affiche un fichier texte.

Net use

Connecte un partage réseau à une lettre de lecteur. La commande **net use** est disponible, avec d'autres paramètres, depuis l'invite de commande.

Rd (rmdir)

Supprime un répertoire. Ne possède pas les options disponibles dans l'invite de commande.

Ren (rename)

Change le nom d'un fichier.

Set

Affiche et définit les variables d'environnement de la console de récupération. **Des détails sont disponibles dans un chapitre précédent.**

Systemroot

Définit le répertoire comme le répertoire racine du système sur lequel vous êtes actuellement connecté. Non disponible dans l'invite de commande.

Type

Affiche un fichier texte.

XVII - Gérer le Registre Windows à partir de l'invite de commande avec « Reg.exe »



Ce chapitre est une traduction de la page  <http://commandwindows.com/reg.htm> dont l'auteur original est Victor Laurie.

L'outil en ligne de commande *reg.exe* est un moyen utile pour gérer le registre de Windows. Ce chapitre décrit ses fonctionnalités et ses utilisations.

Beaucoup d'entre vous doivent connaître l'outil avec interface graphique *regedit.exe* qui permet de modifier le Registre de Windows. Bien moins connu, l'utilitaire en ligne de commande *reg.exe* est aussi fourni avec Windows. Cet accessoire peut faire les mêmes choses que fait *regedit.exe* et présente l'avantage d'être directement utilisable dans les scripts. Il s'agit d'un outil couramment utilisé par les administrateurs système qui gèrent beaucoup d'ordinateurs, mais il peut également être utile aux utilisateurs de PC expérimentés. Je vais décrire ici quelques aspects qui sont intéressants pour ces derniers. Vous trouverez plus de détails sur ce [site Microsoft](#). Il y a également des informations dans le *Centre d'aide et de support*.

La modification du registre ne s'adresse pas à tout le monde, mais il s'agit d'une opération moins redoutable qu'il n'y paraît. Veillez simplement à suivre la règle d'or de sauvegarde du registre avant toute modification. Il y a beaucoup d'astuces qui impliquent une modification du registre simple et « *reg.exe* » fournit un moyen qui est, d'une certaine manière, plus simple et plus sûr que Regedit. Cet outil permet de sauvegarder des clés ou des ruches du Registre dans des fichiers qui peuvent être stockés en dehors du disque principal.

Comme quelques autres utilitaires en ligne de commande, la commande « *reg* » est une console qui possède sa propre série de sous-commandes. Une commande complète va ressembler à

```
reg subcommand variables
```

Le tableau I liste ces sous-commandes et certaines sont détaillées dans les paragraphes suivants. Les commandes peuvent être utilisées sur des ordinateurs distants sur le réseau comme pour l'ordinateur local, mais je vais me limiter aux explications des opérations concernant l'ordinateur local uniquement.

Tableau I. Sous-commandes de *reg.exe*

Sous-commande	Fonction
add	Ajoute une nouvelle sous-clé ou valeur au Registre
delete	Supprime une sous-clé ou des valeurs du Registre
query	Affiche les données dans une sous-clé ou une valeur
compare	Compare les sous-clés ou valeurs spécifiées
copy	Copie une sous-clé dans une autre sous-clé
save	Enregistre une copie des sous-clés, entrées ou données spécifiées du registre dans un format binaire
restore	Réécrit les sous-clés et valeurs sauvegardées au format binaire dans le Registre
load	Réécrit les sous-clés et valeurs sauvegardées au format binaire dans une autre sous-clé
unload	Décharge une section du registre qui a été chargée avec reg load
export	Crée une copy des sous-clés, valeurs ou données spécifiées dans un fichier au format REG (texte)
import	Fusionne un fichier REG contenant des sous-clés ou valeurs dans le registre

XVII-A - Reg add

Cette commande est utilisée pour ajouter des clés et des valeurs dans le registre. La syntaxe est :

```
REG ADD Nom_de_clé [/v Nom_de_valeur | /ve] [/t Type] [/s Séparateur] [/d Données] [/f]
```

Le tableau II explique les paramètres.

Tableau II. Paramètres de la commande REG ADD

Paramètre	Description
KeyName	Nom complet d'une clé de Registre Utilise les abréviations HKCR, HKCU, HKLM et HKU pour les clés racines
/v Nom_de_valeur	Ajoute ou change une valeur
/ve	Change la valeur par défaut de la clé
/t Type	Le type de valeur : REG_BINARY, REG_DWORD, REG_SZ, REG_MULTI_SZ, etc. Si ce paramètre est omis, REG_SZ est pris par défaut.
/s Séparateur	Spécifie le caractère à utiliser comme séparateur dans votre chaîne de données

	pour REG_MULTI_SZ. Si ce paramètre est omis, utilise « \0 » comme séparateur.
/d Données	Données à affecter au Nom_de_valeur ajouté
/f	Force l'écrasement de l'entrée de Registre existante, sans confirmation.

REG ADD fournit un moyen simple et rapide pour ajouter de nouvelles clés dans le Registre ou pour modifier les clés existantes. Nous allons par exemple regarder comment ajouter la sous-clé « HackersAreUs » à la clé Local Machine Software. La commande sera

```
REG ADD HKLM\Software\HackersAreUs
```

Ajoutons maintenant une valeur nommée « Stuff » de type binaire contenant la donnée « 0001 ». La commande sera

```
REG ADD HKLM\Software\HackersAreUs /v Stuff /t REG_BINARY /d 0001
```

Les deux commandes auraient pu être fusionnées en une seule, mais je les ai séparées pour que le processus soit plus clair. J'ai utilisé des majuscules pour REG ADD, mais ce n'est que pour la clarté, elles ne sont pas nécessaires.

XVII-B - Reg delete

Les clés et les valeurs peuvent être supprimées d'une manière similaire, mais plus simple. La syntaxe est :

```
REG DELETE Nom_de_clé [/v Nom_de_valeur | /ve | /va] [/f]
```

Le tableau III détaille les paramètres.

Tableau III. Paramètres de la commande REG DELETE

Paramètre	Description
KeyName	Nom complet d'une clé de Registre Utilise les abréviations HKCR, HKCU, HKLM et HKU pour les clés racines
/v Nom_de_valeur	Supprime une valeur
/ve	Supprime la valeur du nom de valeur vide (par défaut)
/va	Supprime toutes les valeurs d'une clé
/f	Force la suppression sans demander de confirmation

XVII-C - Sauvegarder et restaurer le Registre

L'utilisation la plus importante de *regedit.exe* est la sauvegarde et la restauration du Registre. Deux formats peuvent être utilisés, soit le format binaire connu sous le nom de *ruche* (hive) ou un format texte spécial connu sous le nom de fichier REG. Le dernier format est le plus connu, car il est souvent utilisé pour les astuces du Registre. Les commandes correspondantes sont décrites dans les paragraphes suivants.

XVII-C-1 - Sauvegarder et restaurer des fichiers ruches (hives)

Pour créer une sauvegarde binaire, utilisez la commande

```
REG SAVE Nom_de_clé Nom_de_fichier
```

Les fichiers rches sont préférables pour une sauvegarde, car ils remplacent complètement le contenu des clés quand ils sont restaurés. La commande de restauration est :

```
REG RESTORE Nom_de_clé Nom_de_fichier
```

XVII-C-2 - Exporter et importer des fichiers REG

Les fichiers REG sont des fichiers textes utilisant un format spécial avec l'extension « .reg ». Ce sont des copies d'une ou plusieurs clés du registre. Ils sont souvent utilisés comme un moyen pour transporter de petites modifications du registre. Ils peuvent aussi être utilisés pour effectuer une sauvegarde. Les commandes sont

```
REG EXPORT Nom_de_clé Nom_de_fichier
```

et

```
REG IMPORT Nom_fichier
```

Remarquez que quand un fichier REG est importé, il est *fusionné* avec les entrées du registre au lieu de les remplacer complètement. Les valeurs qui ne sont pas présentes dans le fichier REG ne sont pas supprimées.

XVII-D - Reg Query

Si vous voulez consulter rapidement le contenu d'une clé de registre particulière ou d'une valeur particulière, vous pouvez utiliser la commande

```
REG QUERY Nom_de_clé [/v nom_de_valeur | /ve] [/s]
```

Le seul nouveau paramètre ici est « /s ». Ce commutateur va rechercher toutes les sous-clés et toutes les valeurs de façon récursive (identique à dir /s).

XVIII - Gérer les Services Windows avec la commande de contrôle des services SC



Ce chapitre est une traduction de la page <http://commandwindows.com/sc.htm> dont l'auteur original est Victor Laurie.

L'utilitaire de contrôle des services SC est un puissant outil en ligne de commande permettant de gérer les services Windows. Ce chapitre décrit ses différentes fonctionnalités et possibilités.

Beaucoup de processus, de fonctionnalités du système d'exploitation Windows et d'autres logiciels sont classés dans la rubrique « services ». La gestion des services avec l'outil graphique est décrite [sur un autre site](#). En plus de l'interface graphique de gestion des services, Windows XP dispose d'un puissant utilitaire en ligne de commande. Cet utilitaire, le contrôleur de services, est accessible en tapant « sc » dans l'invite de commande.

Cette méthode de gestion en ligne de commande des services présente l'avantage de pouvoir être utilisée dans les scripts. Cet outil permet également d'arrêter et de démarrer rapidement les services afin de résoudre les problèmes. Les administrateurs système l'utilisent pour gérer les services sur le réseau et pour mettre en place des configurations détaillées. Pour les utilisateurs standard, cette commande fournit un moyen simple de démarrer ou arrêter les services pour voir comment les performances en sont affectées.

XVIII-A - Les sous-commandes de SC

La commande « sc » est fournie avec un certain nombre de sous-commandes. Vous pouvez en consulter la liste sur cette [page Microsoft](#) ou en saisissant « sc /? » dans une invite de commande. Vous trouverez également une liste dans le *Centre d'aide et de support* de Windows XP. La totalité des 24 sous-commandes sont listées. Chaque sous-commande possède un sous-ensemble de commandes. Le tableau ci-dessous liste une sélection des sous-commandes les plus utiles pour les utilisateurs typiques et de leur fonction. Des informations plus détaillées sont disponibles dans le *Centre d'aide et de support* en cherchant « sc ».

Tableau I. Sous-commandes de SC

Commande	Fonction
sc config	Configure le démarrage du service et les comptes d'utilisateur
sc continue	Réactive un service en pause
sc enumdepend	Liste les services dépendants
sc failure	Spécifie l'action à effectuer en cas d'échec d'exécution du service
sc pause	Met un service en pause
sc qc	Affiche la configuration d'un service en particulier
sc query	Affiche des informations sur le service, pilote, type de service ou type de pilote spécifié
sc start	Démarré un service
sc stop	Envoie une requête STOP à un service (il risque de ne pas répondre)

XVIII-B - Exemples d'applications utiles de SC

Les commandes disponibles sont très utiles et permettent une configuration avancée des services. Bien que toutes les fonctionnalités ne soient pas intéressantes pour tous les utilisateurs, quelques-unes sont utiles pour une utilisation

au quotidien. Vous pouvez savoir si un service est démarré, vous pouvez l'arrêter, le démarrer, ou le mettre en pause. Vous pouvez aussi déterminer s'il sera lancé au démarrage du système. Je vais décrire ici quelques-unes des commandes qui me semblent intéressantes.

sc config

Cette commande a de nombreuses fonctions, mais l'une d'elles est de déterminer le statut d'un service au démarrage du système. Un service peut être configuré pour démarrer automatiquement, manuellement ou ne pas démarrer du tout. La syntaxe est :

```
sc config NomDuService start= option
```

Ici, *NomDuService* est le nom du service à configurer et *option* peut prendre l'une des valeurs suivantes : *auto*, *demand*, ou *disabled*. Par exemple pour configurer un service pour qu'il démarre manuellement, la commande à utiliser est :

```
sc config NomDuService start= demand
```

Remarquez qu'il faut une espace après le signe égal (=). La valeur correcte pour le paramètre *NomDuService* n'est pas toujours évidente à connaître et la commande suivante peut être utilisée pour trouver tous les services.

sc query

Cette commande permet d'obtenir des informations sur les services et drivers. Utilisée sans argument, elle renvoie la liste des services en cours d'exécution avec plusieurs informations pour chaque service. La liste n'est pas pratique à lire sur un écran et peut être redirigée dans un fichier texte. Pour créer un fichier texte contenant la liste des services en cours d'exécution, utilisez la commande

```
sc query > listeServices.txt
```

Vous pouvez définir le chemin qui vous convient pour le fichier *listeServices.txt*. Pour créer une liste de tous les services, utilisez

```
sc query type= service state= all > listeTousLesServices.txt
```

Pour créer une liste des drivers actifs, utilisez

```
sc query type= driver
```

ou pour une liste exhaustive de tous les objets :

```
sc query state= all
```

sc start

Pour démarrer un service qui n'est pas en cours d'exécution, utilisez

```
sc start NomDuService
```

sc stop

Pour arrêter un service en cours d'exécution, utilisez

```
sc stop NomDuService
```

Toutefois, certains services ne peuvent pas ou ne devraient pas être arrêtés.

XIX - Gérer les programmes de Windows à partir de l'invite de commande : Tasklist



Ce chapitre est une traduction de la page <http://commandwindows.com/tasklist.htm> dont l'auteur original est Victor Laurie.

Windows XP est fourni avec un outil puissant en ligne de commande appelé Tasklist qui fournit des détails sur les programmes et les processus en cours d'exécution.

Beaucoup d'utilisateurs connaissent le Gestionnaire de tâches graphique, que j'ai décrit sur [un autre site](#), et qui fournit plusieurs sortes d'informations sur les applications et les processus qui s'exécutent sur le système. Il existe également plusieurs outils en ligne de commande qui fournissent les mêmes informations, mais de manière plus détaillée. Dans ce chapitre, je vais décrire les fonctionnalités de l'outil *Tasklist* (le fichier système est : *tasklist.exe*). Cet outil est inclus dans l'installation de la version professionnelle de XP, mais n'est pas fourni dans la version familiale. Cependant, les utilisateurs de la version familiale de XP peuvent [télécharger tasklist ici](#). Tasklist peut être utilisé pour savoir combien de mémoire et de temps CPU les processus utilisent, quelles DLL sont utilisées, et bien d'autres informations. De plus il peut être très utile pour résoudre les problèmes.

XIX-A - Commande Tasklist basique

Si vous voulez simplement savoir les tâches en cours d'exécution, entrez TASKLIST en ligne de commande. Vous pouvez rediriger le résultat de la commande dans un fichier si vous le souhaitez. Le format par défaut est un tableau comportant plusieurs colonnes d'information. La figure ci-dessous montre un exemple partiel de résultat obtenu. Il y a cinq colonnes d'information. La liste suivante donne la signification des différents en-têtes des colonnes.

Nom de l'image

Le nom du processus ou du fichier exécutable exécutant le processus.

PID

L'identifiant de processus. Le système assigne un numéro à chaque processus pour pouvoir en garder la trace. Il est possible d'avoir plusieurs processus en cours d'exécution portant le même nom, mais le PID sera unique pour chacun. Notez que le PID ne sera pas le même à chaque fois que vous ouvrirez un programme en particulier. Vous pouvez avoir besoin du PID pour utiliser certains autres outils de diagnostic, et Tasklist vous fournit un moyen d'obtenir cette information.

Nom de la session

À moins que vous ne soyez sur un réseau, vous devriez lire « Console » indiquant que le processus a été démarré localement. Les utilisateurs d'ordinateurs personnels peuvent généralement ignorer cette colonne.

Numéro de session

Chaque session possède un numéro assigné. Les utilisateurs d'ordinateurs personnels peuvent ignorer également cette colonne.

Utilisation mémoire

Cette colonne donne des informations utiles pour savoir combien de mémoire (en Ko) un processus utilisait au moment où Tasklist a été exécuté.

```
C:\>tasklist
```

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Console	0	16 K
System	4	Console	0	212 K
smss.exe	520	Console	0	464 K
csrss.exe	584	Console	0	1,160 K
winlogon.exe	608	Console	0	4,732 K
services.exe	652	Console	0	3,184 K
lsass.exe	664	Console	0	1,380 K
svchost.exe	824	Console	0	3,216 K
svchost.exe	888	Console	0	18,020 K
svchost.exe	988	Console	0	1,964 K
svchost.exe	1016	Console	0	3,116 K
spoolsv.exe	1048	Console	0	4,196 K
Explorer.EXE	1596	Console	0	4,740 K
avgamsvr.exe	1692	Console	0	4,476 K
avgupsvc.exe	1772	Console	0	2,312 K
wdfmgr.exe	1852	Console	0	1,564 K
usmon.exe	1884	Console	0	9,052 K
OneTouch.exe	112	Console	0	2,264 K

Des colonnes supplémentaires peuvent être affichées en mode « verbose » qui est obtenu en utilisant le commutateur « /v ». Les colonnes sont les suivantes :

Status

Donne l'état du processus : « Running », « Not Responding », ou « Unknown ». Utile pour trouver les processus plantés. Le statut « Unknown » (inconnu) peut faire référence à un processus normal, mais « Not Responding » (pas de réponse) indique un processus qui devrait être arrêté.

Nom d'utilisateur

Indique le nom du compte qui a lancé le processus, Windows lui-même va démarrer de nombreux processus et les comptes systèmes qui vont apparaître dans cette colonne sont : SYSTEM, SERVICE LOCAL ou SERVICE RÉSEAU, accompagnés du nom de domaine local AUTORITE NT.

Temps processeur

La quantité *totale* de temps processeur utilisé par le processus depuis son démarrage. Ce nombre peut être très grand si vous n'éteignez jamais l'ordinateur.

Titre de la fenêtre

Nom de la fenêtre du processus, si elle existe. Aide parfois à identifier quel programme est en cause.

XIX-B - Plus d'options avancées pour Tasklist

Il y a beaucoup d'options qui sont fournies par des commutateurs. La syntaxe complète est :

```
TASKLIST [/S système [/U utilisateur [/P mot_de_passe]]] [/M [module] | /SVC | /V] [/FI filtre]
[/FO format] [/NH]
```

Les majuscules ne sont utilisées que par souci de clarté, mais la commande est insensible à la casse. Le tableau I décrit les différents paramètres.

Tableau I. Paramètres de TASKLIST

Paramètre	Description
/S système	Spécifie le système distant auquel se connecter. N'est pas nécessaire pour l'ordinateur local
/U utilisateur	Spécifie le contexte utilisateur sous lequel la commande doit exécuter. N'est pas nécessaire pour l'ordinateur local
/P [mot_de_passe]	Spécifie le mot de passe pour le contexte utilisateur donné. Il est demandé s'il est omis.
/M [module]	Liste toutes les tâches utilisant le nom de fichier exe ou dll donné. Si le nom de module n'est pas spécifié, tous les modules chargés sont affichés.
/SVC	Affiche les services hébergés dans chaque processus.
/V	Affiche les informations de tâches détaillées.
/FI filtre	Affiche un ensemble de tâches qui correspond au critère spécifié par le filtre.
/FO format	Spécifie le format de sortie. Valeurs valides : « TABLE », « LIST », « CSV ».
/NH	Spécifie que les en-têtes de colonnes ne doivent pas être affichés sur la sortie. Valide uniquement pour les formats « TABLE » et « CSV ».

Ces paramètres additionnels permettent à Tasklist de fournir des informations très détaillées sur le système. Quelques exemples sont expliqués dans les paragraphes suivants.

XIX-B-1 - Trouver quels services utilisent un processus

Il peut être utile de connaître la relation entre un processus et les services qui s'exécutent sur un système (pour un article sur les *services*, [consultez cette page](#)). Pour obtenir un tableau associant le nom de l'image, le PID et les services, utilisez la commande

```
tasklist /svc >liste.txt
```

J'ai montré ici la redirection vers un fichier pour illustrer la création d'un rapport texte. Cette commande peut être utilisée pour diagnostiquer des problèmes avec un service en surveillant l'utilisation de la mémoire et d'autres propriétés des processus associés avec le service.

XIX-B-2 - Trouver quels fichiers DLL sont utilisés par un processus

Les processus peuvent utiliser différents fichiers DLL en appelant diverses procédures depuis leurs bibliothèques. Il n'est pas rare qu'un problème survienne en raison d'une DLL endommagée ou qui n'est pas dans la bonne version. Pour trouver quelles DLL sont utilisées par chaque processus, utilisez la commande

```
tasklist /m
```

Cette commande va renvoyer un tableau reliant le nom de l'image, le PID et les modules. Ici, les « modules » indiquent les DLL. La table peut contenir beaucoup d'entrées, mais vous pouvez limiter la liste à une DLL spécifique en utilisant son nom dans la commande. Par exemple, pour voir uniquement les processus qui utilisent *oleaut32.dll*, tapez

```
tasklist /m oleaut32.dll
```

XIX-C - Filtrer le résultat de Tasklist

La sortie peut être réduite à des paramètres spécifiques en utilisant des filtres et le commutateur */FI*. Il existe de nombreux opérateurs de comparaison, ils sont détaillés dans le tableau II. Tous les opérateurs ne peuvent pas être utilisés avec tous les paramètres et les valeurs autorisées sont listées dans le tableau III pour les paramètres les plus utiles.

Tableau II. Opérateurs de comparaison pour les filtres

Opérateur	Description
eq	Egal (Equal en anglais)
ne	Est différent de (Not Equal)
gt	Plus grand que (Greater Than). Utilisé uniquement avec des valeurs numériques
lt	Plus petit que (Less Than). Utilisé uniquement avec des valeurs numériques
ge	Supérieur ou égale à (Greater or Equals) Utilisé uniquement avec des valeurs numériques
le	Inférieur ou égale (Less or Equal) Utilisé uniquement avec des valeurs numériques

Tableau III. Opérateurs des filtres et valeurs autorisées

Paramètre	Opérateurs valides	Valeurs valides
ImageName	eq, ne	Chaine de caractères valide
PID	eq, ne, gt, lt, ge, le	Entier positif valide
MemUsage	eq, ne, gt, lt, ge, le	Entier positif valide en kilo-octets
Status	eq, ne	Running, Not Responding, Unknown
Username	eq, ne	Un nom d'utilisateur valide (peut être SYSTEM, SERVICE LOCAL, SERVICE RÉSEAU)
WindowTitle	eq, ne	Chaine de caractères valide

Voici un exemple d'utilisation des filtres avec une commande permettant de trouver les processus qui ne répondent pas. La commande sera

```
tasklist /fi "status eq not responding"
```

Un autre exemple est la recherche des processus qui utilisent beaucoup de mémoire, par exemple plus de 40 Mo. La commande est :

```
tasklist /fi "memusage gt 40000"
```

Ce dernier exemple montre comment clarifier les multiples occurrences du processus « svchost.exe ». (Chacun a un PID différent). Service Host (svchost.exe) est un élément de base du système d'exploitation Windows qui est utilisé dans beaucoup de systèmes de bas niveau. Ces derniers sont placés dans plusieurs groupes de services, tous s'exécutant sous le nom de service générique « svchost.exe ». (**Vous pouvez consulter cette explication**). Pour voir quels services sont associés à chaque instance de svchost.exe, utilisez la commande

```
tasklist /svc /fi "imagename eq svchost.exe"
```

Vous trouverez plus d'informations sur Tasklist sur ce [site Microsoft](#).

XX - Gérer les programmes de Windows : Tskill et Taskkill



Ce chapitre est une traduction de la page <http://commandwindows.com/taskkill.htm> dont l'auteur original est Victor Laurie.

Windows XP est fourni avec plusieurs outils pour terminer les programmes ou les processus à partir de l'invite de commande. Ce chapitre décrit les fonctionnalités et les utilisations de Taskkill et Tskill.

Il est parfois souhaitable de terminer un programme ou un processus à partir de la ligne de commande. Le processus peut être planté ou ne pas répondre ou vous souhaitez peut-être avoir un script pour le terminer. Les deux versions de Windows XP (Familiale et Professionnelle) sont fournies avec l'outil **Tskill**. XP Professionnel dispose en plus d'un outil plus puissant : **Taskkill**. Bien que l'interface graphique du **Gestionnaire de Tâches** puisse être utilisée pour terminer les programmes qui sont plantés, la ligne de commande peut être plus rapide et plus facile à utiliser. De même, dans certaines situations, il peut être pratique d'avoir un fichier batch qui se lance comme un script. De plus, Taskkill est capable d'utiliser des filtres sophistiqués.

XX-A - Tskill

La syntaxe de cette commande est :

```
TSKILL IDprocessus | NomProcessus [/SERVER:NomServeur] [/ID:IDsession | /A] [/V]
```

La signification des différents paramètres est donnée dans le tableau I.

Tableau I. Paramètres de la commande Tskill

Paramètre	Description
processid	PID du processus à terminer. Utiliser uniquement si <i>processname</i> n'est pas utilisé
processname	Nom du processus à terminer. Les jokers peuvent être utilisés ici. Ne pas utiliser si le PID est utilisé.
/SERVER:NomServeur	Serveur contenant l'ID de processus (ID actuel par défaut). Généralement pas nécessaire sur les PC domestiques.
/ID:IDsession	Arrêt du processus exécuté au cours de la session spécifiée. Souvent inutilisé sur les PC domestiques.
/A	Arrêt du processus exécuté au cours de TOUTES les sessions (nécessite les privilèges d'administration).
/V	Affiche les informations sur les actions exécutées.

Voici un exemple d'une commande simple pour terminer Notepad :

```
tskill notepad
```

Cet autre exemple termine tous les documents Microsoft que vous avez ouvert :

```
tskill winword
```

Tous les documents Word ouverts seront fermés, mais le contenu ne sera pas enregistré donc commencez par sauvegarder vos documents importants. Un administrateur peut terminer les processus qui s'exécutent dans des sessions démarrées par d'autres utilisateurs. La commande

```
tskill winword /a
```

va fermer les documents Word de tous les utilisateurs.

Il n'est pas toujours évident de savoir quel nom de processus utiliser pour un programme. En général, le nom du fichier exécutable (sans l'extension EXE) fonctionne. Une méthode utilisable est de se servir de **Tasklist** pour trouver le PID et utiliser celui-ci. Une autre méthode est d'utiliser le **Gestionnaire des tâches** pour trouver le processus associé à un programme. (Le Gestionnaire de tâches peut bien sûr être utilisé pour terminer un programme).

XX-B - Taskkill

Taskkill est un outil qui dispose de plus d'options. La syntaxe de la commande est :

```
TASKKILL [/S système] [/U utilisateur [/P [mot_passe]]] { [/FI filtre] [/PID ID_processus | /IM image] } [/F] [/T]
```

Les différents paramètres sont décrits dans le tableau II.

Tableau II. Paramètres de la commande Taskkill

Paramètre	Description
/S système	Spécifie le système distant auquel se connecter. N'est pas nécessaire pour l'ordinateur local
/U utilisateur	Spécifie le contexte utilisateur sous lequel la commande doit s'exécuter. Souvent inutilisé sur les PC domestiques.
/P mot_de_passe	Mot de passe pour l'utilisateur
/FI filtre	Affiche un ensemble de tâches qui correspond au critère spécifié par le filtre.
/PID ID_processus	Spécifie le PID du processus à terminer. Non utilisé quand le nom de l'image est donné dans la commande
/IM image	Spécifie le nom de l'image du processus à terminer. Le joker « * » peut être utilisé pour spécifier tous les noms d'image. Non utilisé si le PID est donné dans la commande
/F	Force la fermeture de tous les processus
/T	Met fin au processus spécifié et tous les processus enfant qu'il a démarrés.

Les paramètres comme le nom de l'image ou le PID peuvent ne pas être immédiatement évidents à trouver, mais **Tasklist** peut être utilisé pour les obtenir. Taskkill a plus d'options que Tskill et est donc plus compliqué à utiliser. Par exemple la simple commande « Taskkill notepad » ne va pas fonctionner. D'abord, le nom de l'image est « notepad.exe » et non pas le nom du programme « notepad ». De plus, vous allez généralement devoir utiliser le commutateur de forçage /F. La commande pour fermer notepad sera alors

```
taskkill /im notepad.exe /f
```

Cet autre exemple montre comment fermer plusieurs programmes en une seule commande :

```
taskkill /f /im notepad.exe /im mspaint.exe
```

La documentation n'est pas précise sur le fait d'utiliser le commutateur `/f` avant ou après le nom de l'image, mais il semble que cela n'a pas d'importance.

XX-C - Filtrer le résultat de Taskkill

Taskkill devient particulièrement puissant quand on utilise les filtres avec le commutateur « `/fi` ». Diverses règles peuvent être formées à l'aide des opérateurs de comparaison indiqués dans le tableau III.

Tableau III. Opérateurs de comparaison pour les filtres

Opérateur	Description
eq	Egal (equals)
ne	Est différent de (Not equals)
gt	Plus grand que (Greater Than). Utilisé uniquement avec des valeurs numériques
lt	Plus petit que (Less Than). Utilisé uniquement avec des valeurs numériques
ge	Supérieur ou égale à (Greater or Equals). Utilisé uniquement avec des valeurs numériques
le	Inférieur ou égale (Less or Equal). Utilisé uniquement avec des valeurs numériques

Le tableau IV indique les variables qui peuvent être utilisées dans un filtre.

Tableau IV. Opérateurs des filtres et valeurs autorisées

Paramètre	Opérateurs valides	Valeurs valides
ImageName	eq, ne	Chaine de caractères valide
PID	eq, ne, gt, lt, ge, le	Entier positif valide
MemUsage	eq, ne, gt, lt, ge, le	Entier positif valide en kilo-octets
CPUTime	eq, ne, gt, lt, ge, le	Temps processeur au format <i>hh:mm:ss</i> .
Session	eq, ne, gt, lt, ge, le	Numéro de session
Status	eq, ne	Running, Not Responding
Username	eq, ne	Un nom d'utilisateur valide (peut être SYSTEM, SERVICE LOCAL, SERVICE RÉSEAU)
WindowTitle	eq, ne	Chaine de caractères valide
Services	eq, ne	Nom de service
Modules	eq, ne	Nom de DLL

XX-D - Exemple d'utilisation des filtres dans Taskkill

Avec les filtres, vous pouvez imposer un ensemble de conditions qui doivent être respectées. Les filtres donnent à Taskkill une souplesse considérable et vous permettent de définir finement la cible. Quelques exemples sont donnés ci-dessous. Remarquez qu'il n'est pas nécessaire d'utiliser un nom d'image spécifique ou un PID quand vous utilisez les filtres.

Forcer la fermeture de tous les processus qui ne répondent pas. Peut être utilisé pour créer un petit fichier batch qui termine les programmes plantés ou figés.

```
taskkill /f /fi "status eq not responding"
```

Forcer la fermeture de tous les programmes utilisant un fichier DLL spécifique appelé « some.dll ». Cela doit être utilisé avec prudence, mais une utilisation peut être de terminer les processus associés avec une DLL provenant d'un spyware ou d'un cheval de Troie. Utilisez **Tasklist** pour voir quels sont les programmes qui utilisent une DLL donnée.

```
taskkill /f /fi "modules eq some.dll"
```

Fermer tous les programmes qui utilisent une grande quantité de mémoire, par exemple > à 40 Mo. Utiliser avec précaution.

```
taskkill /f /fi "memusage gt 40000"
```

Fermer tous les programmes utilisant plus de 40 Mo de mémoire, mais pas l'explorateur Windows.

```
taskkill /f /fi "imagename ne explorer.exe" /fi "memusage gt 40000"
```

XXI - Exécuter des fichiers VBScripts et JScript à partir de l'invite de commande



Ce chapitre est une traduction de la page <http://commandwindows.com/scripts.htm> dont l'auteur original est Victor Laurie.

Ce chapitre décrit l'utilisation de l'hôte des scripts Windows pour exécuter des scripts dans l'invite de commande. Des exemples spécifiques sont également donnés.

Windows XP est fourni avec deux interfaces pour exécuter des fichiers VBScript et JScript (version Microsoft de JavaScript). L'interface par défaut est une interface graphique utilisant *Wscript.exe*. Cependant, il y a aussi une interface en ligne de commande *Cscript.exe*. Il est souvent plus pratique d'utiliser la ligne de commande pour les scripts et Windows XP est fourni avec quelques scripts utiles pour l'administration système. Ces scripts doivent être lancés depuis l'invite de commande.

XXI-A - Configurer l'invite de commande pour exécuter des scripts

Beaucoup de scripts pour la gestion de l'ordinateur sont plus pratiques à utiliser à partir de l'invite de commande de telle sorte que les utilisateurs n'ont pas à rencontrer les complications des fenêtres et des boîtes de dialogue de l'interface graphique. Il est même plus facile de manipuler le résultat dans une fenêtre de commande. Afin d'exécuter un script en ligne de commande, il est possible de préfixer le nom du script avec le nom de l'exécutable *Cscript.exe*. Par exemple, une commande de la forme

```
cscript.exe unScript.vbs
```

peut être saisie en ligne de commande et *unScript.vbs* sera alors exécuté dans la fenêtre de commande. Remarquez toutefois qu'il faut généralement utiliser le chemin complet, et les chemins contenant des espaces doivent être entourés de guillemets. Pour faire du processeur de script Cscript l'hôte de scripts par défaut, tapez dans une ligne de commande

```
wscript //H:cscript
```

Les scripts peuvent ensuite être lancés en entrant simplement en ligne de commande

```
unScript.vbs
```

Pour revenir à la valeur par défaut avec interface graphique, utilisez la commande

```
wscript //H:wscript
```

XXI-B - Quelques scripts fournis avec Windows XP

Cachés dans le dossier **Windows\System32** se trouvent certains fichiers au format VBScript dont la majeure partie des utilisateurs n'a jamais entendu parler. Certains outils inclus avec les **packages d'outils système** décrits sur une **autre page** sont sous la forme de scripts. Certains ne s'adressent qu'aux administrateurs, mais plusieurs d'entre eux peuvent être utiles à la majorité des propriétaires d'ordinateurs personnels. Ils sont listés ci-dessous. Vous trouverez plus de détails sur les scripts dans le **Centre d'aide et de support** de Windows XP. Il est plus facile de lancer ces scripts si Cscript est configuré comme interface par défaut.

Eventquery.vbs

Liste les événements et les propriétés d'événement à partir d'un ou plusieurs journaux d'événements. Peut être utilisé avec un filtre pour spécifier les types d'événements à inclure ou à exclure de la recherche.

Pagefileconfig.vbs

Permet à un administrateur d'afficher et configurer les réglages de pagination de la Mémoire Virtuelle.

Prncnfg.vbs

Configure ou affiche les informations de configuration d'une imprimante. Utilisé sans paramètre, *prncnfg.vbs* affiche l'aide de ligne de commande.

Prndrvr.vbs

Ajoute, supprime ou liste les pilotes d'impression. Utilisé sans paramètre, *prndrvr.vbs* affiche l'aide de ligne de commande.

Prnjobs.vbs

Met en pause, réactive, et liste les travaux d'impression. Utilisé sans paramètre, *prnjobs.vbs* affiche l'aide de ligne de commande.

Prnmngr.vbs

Ajoute, supprime ou liste les imprimantes ou connexions d'imprimante, affiche et règle l'imprimante par défaut. Utilisé sans paramètre, *prncnfg.vbs* affiche l'aide de ligne de commande.

Prnport.vbs

Crée, supprime et liste les ports d'imprimantes standards TCP/IP, affiche et change la configuration des ports. Utilisé sans paramètre, *prnport.vbs* affiche l'aide de ligne de commande.

Prnqctl.vbs

Imprime une page de test, met en pause ou réactive une imprimante, et supprime une file d'impressions. Utilisé sans paramètre, *prnqctl.vbs* affiche l'aide de ligne de commande.

XXI-C - Outil de ligne de commande WMI : WMIC (Windows Management Instrumentation Command-line)

WMIC est une ligne de commande et une interface de script qui simplifie l'utilisation de WMI (Windows Management Instrumentation). WMIC est basé sur les alias. Les alias rendent disponibles les informations fournies par WMI sans avoir à comprendre les concepts spécifiques à WMI. Vous trouverez plus de détails sur cette [référence Microsoft](#). Des informations sont également disponibles sur un ordinateur local en tapant dans l'invite de commande

```
WMIC /?
```

WMIC peut être utilisé pour écrire des scripts simples pour automatiser la gestion de l'ordinateur.

XXII - Outils utiles pour Windows XP provenant du Resource Kit de Windows Server 2003



Ce chapitre est une traduction de la page <http://commandwindows.com/server2003tools.htm> dont l'auteur original est Victor Laurie.

Plusieurs outils en ligne de commande du kit de ressources de Windows Server 2003 peuvent être utilisés dans Windows XP. Certains sont décrits dans ce chapitre.

Microsoft fournit un grand assortiment d'outils en ligne de commande. Ceux-ci peuvent être obtenus de plusieurs façons. Pour les utilisateurs de Windows XP Professionnel, il existe trois sources principales. Un ensemble d'outils « natifs » sont inclus dans l'installation standard et sont décrits sur [une autre page](#). Un second ensemble appelé « support tools », peut être téléchargé, ou installé à partir d'une version complète de Windows XP Professionnel. Ils sont décrits sur [cette page](#). Un troisième lot fait partie du Kit de Ressources de Windows Server 2003, mais est utilisable sur Windows XP. Ils peuvent être [téléchargés ici](#) et sont décrits dans ce chapitre.

Comme on pouvait s'y attendre, les outils de Windows Server 2003 sont des outils destinés à des systèmes utilisant de grands réseaux, et nombreux sont les outils qui ne sont pas appropriés pour les ordinateurs domestiques. Néanmoins, certains peuvent être utiles dans un système simple. J'ai effectué une sélection arbitraire de ceux qui me semblent les plus utiles et je les décris ci-dessous. Notez que certains de ces outils sont des outils graphiques ouverts à partir de la ligne de commande. Ils seront indiqués quand ce sera le cas. Une liste complète des outils est disponible sur le [site de téléchargement](#) de Microsoft. Une fois que les outils sont installés, le Centre d'aide et de Support va en contenir la liste avec les descriptions et leur syntaxe d'utilisation.

Cdburn.exe : Outil de gravure d'ISO sur CD-ROM

CDBurn est un outil en ligne de commande qui permet à l'utilisateur de graver des images ISO stockées sur le disque dur sur un CD. L'image de données peut être de n'importe quel type de données, même des données brutes. Cet outil peut également être utilisé pour effacer un CD-RW.

ChkLnks.exe : Assistant de contrôle de lien

L'assistant de contrôle des liens (ChkLnks) est un outil graphique qui vérifie tous les fichiers liens (raccourcis) sur un ordinateur pour déterminer si le raccourci pointe sur une application ou un document existant. Quand l'assistant de contrôle des liens ne trouve pas d'application associée ou de document, l'assistant liste que le raccourci est un lien mort, et vous fournit une option pour l'enlever.

Cleanspl.exe : Nettoyeur de Spooler

Le nettoyeur de Spooler (CleanSpl) est un outil graphique qui restaure le spooler d'impression dans son état original. Il supprime tous les travaux d'impression, les imprimantes, les pilotes d'impressions et les fichiers spool sur un serveur local ou distant spécifié.

Createfil.exe Créateur de fichier

CreatFil crée un fichier vide d'une taille spécifiée, rempli de caractères espace.

Diskraid.exe : Outil de configuration RAID

DiskRaid est un outil en ligne de commande qui permet la configuration et la gestion de disques RAID (redundant array of independent (ou inexpensive)).

Dvdburn : Outil de gravure d'ISO sur DVD

DVDBurn est un outil en ligne de commande qui permet à l'utilisateur de graver des images DVD à partir de fichiers images situés sur le disque dur, sur un disque DVD. L'image de données peut être créée par n'importe quel programme capable de produire des fichiers images de DVD. Vous pouvez aussi utiliser cet outil pour effacer un DVD.

Empty.exe : Outil de libération d'espace de travail

L'outil de libération d'espace de travail (Empty) est un outil en ligne de commande qui libère l'espace de travail (espace de mémoire physique) d'une tâche spécifiée ou d'un processus, rendant ces pages disponibles pour d'autres processus.

Iniman.exe : outil de manipulation des fichiers d'initialisation (.ini)

IniMan est un outil en ligne de commande qui vous permet d'ajouter, supprimer, modifier ou d'interroger des sections ou des clés dans un fichier « .ini ».

Instsvr.exe : Installateur de service

L'installateur de service (InstSrv) est un outil en ligne de commande qui installe et désinstalle des services exécutables et leur assigne des noms.

Kernrate.exe : Outil de profilage du noyau (kernel)

L'outil de profilage du noyau (KernRate) est un outil en ligne de commande permettant d'établir un profilage CPU. Il surveille les performances et fournit un rapport à l'utilisateur. KernRate établit des rapports sur l'activité CPU du noyau et des processus utilisateurs. Utilisez KernRate pour identifier les processus qui sont à l'origine d'un goulet d'étranglement sur le processeur.

Linkspeed.exe : Vitesse de liaison

Link Speed est un outil en ligne de commande qui affiche la vitesse de connexion entre les ordinateurs d'un réseau

Memmonitor : Moniteur de mémoire

MemMonitor est un outil en ligne de commande qui surveille la mémoire utilisée par un processus.

memtriage.exe : Outil de tri de fuites de ressource

MemTriage est un outil en ligne de commande qui détecte des fuites possibles de ressources sur un système en fonctionnement. MemTriage enregistre les informations sur les processus ou le pool noyau courant et les enregistre dans un fichier journal.

Now.exe : STDOUT Date et heure actuelle

Now lit l'entrée standard (STDIN) et affiche, sur la sortie standard (STDOUT), la date et l'heure suivie par STDIN. Utilisé sans paramètre, Now affiche la date et l'heure actuelle.

Ntimer : Timer de programmes Windows

NTimer est un outil en ligne de commande qui mesure le temps d'exécution d'un programme. NTimer affiche le temps écoulé, le temps en mode utilisateur, et le temps en mode privilégié.

Oh.exe : Handles ouverts

Open Handles est un outil en ligne de commande qui affiche les handles de toutes les fenêtres ouvertes. OH peut également être utilisé pour afficher uniquement les informations d'un processus, type d'objet ou nom d'objet spécifique.

Oleview.exe : Visionneuse d'objets OLE/COM

La visionneuse d'objets OLE/COM est un outil graphique qui vous permet de générer toutes les classes COM (Microsoft Component Object Model) installées sur votre ordinateur.

Pathman.exe : Gestionnaire de PATH

PathMan est un outil en ligne de commande qui ajoute ou supprime des éléments du PATH système ou du PATH de l'utilisateur.

Pmon.exe : Moniteur de ressources des processus

Le moniteur de ressources des processus (PMon) est un outil en ligne de commande qui affiche différentes mesures sur l'utilisation de la mémoire et du processeur par les processus s'exécutant sur le système. Le résultat de PMon apparaît dans la fenêtre de commandes.

Printdriverinfo.exe : Source des pilotes (drivers)

PrintDriverInfo est un outil en ligne de commande utilisé pour collecter des informations concernant les pilotes d'impression, il est principalement utilisé à des fins de support.

Qgrep.exe

Qgrep.exe est un outil en ligne de commande utilisé pour chercher dans un fichier ou une liste de fichiers une chaîne de caractères spécifique ou un pattern et renvoie la ligne correspondante. QGrep permet également de chercher dans de multiples fichiers et sous-répertoires. Qgrep est similaire à l'outil UNIX « grep ».

Remapkey.exe : Outil de modification de la disposition du clavier Windows

RemapKey est un outil graphique qui permet de changer la disposition d'un clavier en changeant les scan codes des touches.

Robocopy.exe : Utilitaire de copie de fichier (Robust File Copy Utility)

Un utilitaire pour maintenir une copie à l'identique une copie d'un dossier et de ses sous-dossiers dans plus d'un emplacement.

Setprinter.exe : Outil de configuration du Spooler

Setprinter est un outil en ligne de commande utilisé pour définir la configuration des imprimantes locales et distantes.

Sleep.exe : Attente dans fichier Batch

Sleep est un outil en ligne de commande qui force l'ordinateur à attendre durant une durée spécifiée. À utiliser dans les fichiers Batch

Splinfo.exe : Informations de spooler d'impression

SplInfo est un outil en ligne de commande qui collecte et affiche les informations du spooler d'impression.

Tail.exe

Tail est un outil en ligne de commande qui affiche les dernières lignes d'un fichier (dont le nombre est choisi par l'utilisateur), par exemple un fichier journal dans une fenêtre console.

XXIII - Support Tools de Windows XP SP2



Ce chapitre est une traduction de la page <http://commandwindows.com/supporttools.htm> dont l'auteur original est Victor Laurie.

Les systèmes Windows sont fournis avec un certain nombre d'outils en ligne de commande. Il en existe d'autres qui peuvent être téléchargés et installés. Un de ces groupes de commande est appelé « support tools » et un certain nombre d'entre elles est listé dans ce chapitre. Ces outils s'adressent à l'origine à l'administration système, mais quelques-uns peuvent également être utilisés pour diagnostiquer et résoudre les problèmes sur les ordinateurs personnels.

La liste ci-dessous ne mentionne pas certains des outils plus spécialisés, tels que ceux pour Active Directory. La **liste complète est ici** tout comme le téléchargement gratuit. Ces outils viennent en complément d'un grand nombre d'utilitaires en ligne de commande qui font partie de l'installation par défaut de Windows XP Professionnel. Ces outils additionnels nécessitent d'être installés.

XXIII-A - Une sélection des Support Tools de Windows XP en ligne de commande

- *acldiag.exe* - Gère les ACL (Access Control Lists)
- *activate.exe* - Activation de Windows
- *apmstat.exe* - Fournit des informations sur les fonctionnalités de gestion d'alimentation avancée (APM : Advanced Power Management).
- *bindiff.exe* - Affiche les différences entre deux fichiers binaires
- *bitsadmin.exe* - Gère le service de transfert intelligent en arrière-plan (BITS : Background Intelligent Transfer Service).
- *browstat.exe* - Outil général d'analyse en mode caractère
- *cabarc.exe* - Permet aux utilisateurs de créer, de consulter et d'extraire des fichiers Windows Cabinet (.CAB).
- *depends.exe* - Permet de déterminer de quelle DLL dépend une application (connu aussi sous le nom de « Dependency Walker »).
- *dhcpcd.exe* - Affiche les serveurs DHCP actifs dans votre sous-réseau.
- *diruse.exe* - Affiche des informations sur la taille des répertoires sur les volumes NTFS.
- *dmdia.exe* - Affiche l'état du système et les informations de configuration concernant le stockage disque.
- *dupfinder.exe* - Recherche les fichiers en double.
- *efsinfo.exe* - Affiche des informations sur les fichiers encryptés avec EFS (Encrypting File System) sur les partitions NTFS.
- *extract.exe* - Un utilitaire qui vous permet d'extraire tous les fichiers ou un fichier spécifique contenu dans un fichier cabinet (.cab).
- *filever.exe* - Affiche des informations sur les versions des fichiers exécutables.
- *ipseccmd.exe* - Configure les stratégies de sécurité IP (IP Sec).
- *memsnap.exe* - Prend une image des ressources mémoires consommées par tous les processus lancés et écrit ces informations dans un fichier journal.
- *msicuu.exe* - Utilitaire de nettoyage de Windows Installer
- *msizap.exe* - Supprime toutes les informations de Windows Installer sur un produit ou sur tous les produits installés sur un ordinateur.
- *netcap.exe* - Surveille les paquets sur un réseau local et écrit les informations dans un fichier journal.
- *netdiag.exe* - Teste la connectivité réseau.
- *netset.exe* - Utilisé pour ajouter supprimer ou changer une configuration réseau.
- *pfmon.exe* - Affiche les erreurs qui se produisent lors de l'exécution d'un processus.
- *pstat.exe* - Fournit des informations sur les processus et les drivers en cours d'exécution sur votre ordinateur.
- *pviewer.exe* - Visualisateur de processus.
- *setx.exe* - Définit les variables d'environnement.
- *showacces.exe* - Permet aux utilisateurs d'examiner les listes de contrôle d'accès (ACLs)
- *timezone.exe* - Utilitaire de mise à jour de l'heure d'été/hiver.
- *whoami.exe* - Affiche le nom de l'utilisateur et l'identifiant de sécurité (SID).

- *windiff.exe* - Affiche les différences entre les fichiers textes ASCII spécifiés ou entre des dossiers contenant des fichiers textes ASCII.
- *xcaccls.exe* - Utilisé pour définir en ligne de commande les options de sécurité du système de fichier accessibles dans l'explorateur de Windows.

XXIV - Les fichiers Batch (scripts) dans Windows



Ce chapitre est une traduction de la page <http://commandwindows.com/batch.htm> dont l'auteur original est Victor Laurie.

Les fichiers batchs ou scripts sont de petits fichiers texte faciles à écrire qui contiennent une série de commandes. Ils sont assez simples pour que les utilisateurs d'ordinateurs personnels puissent en tirer avantage.

Les administrateurs de systèmes sont bien conscients de l'utilité des fichiers batch, mais les utilisateurs de PC ne les connaissent généralement pas ou sont intimidés par l'écriture ou l'exécution d'un script. C'est dommage, car cela signifie que beaucoup passent à côté d'un outil puissant pour la réalisation de routines ou de tâches répétitives. Bien que les fichiers batch puissent être assez sophistiqués et utilisés pour l'administration de systèmes et de réseaux complexes, ils peuvent aussi être très simples et très courts. Dans ce chapitre, je vais décrire les fichiers batch et expliquer quelques exemples simples qui permettent de simplifier les tâches de base.

XXIV-A - Que-est-ce qu'un fichier batch ?

Il s'agit de simples fichiers textes contenant des lignes de commande qui sont exécutées séquentiellement, l'une après l'autre. Ces fichiers utilisent l'extension spéciale « .bat » ou « .cmd ». Les fichiers de ces types sont reconnus et exécutés via une interface (parfois appelée console) fournie par le système et appelée interpréteur de commandes. Dans Windows XP/Vista, l'interpréteur de commandes est le fichier `cmd.exe`. Le grand assortiment des commandes polyvalentes disponibles dans Windows XP / Vista fait des fichiers batch un outil puissant.

Un fichier batch est simplement créé en ouvrant n'importe quel éditeur de texte comme *Notepad*, en entrant quelques lignes contenant des commandes, et en sauvegardant le fichier avec une extension .bat ou .cmd. (L'extension CMD est limitée aux versions de Windows les plus récentes et les fichiers .cmd ne sont pas reconnus sur les systèmes Windows 9x/Me. Dans Windows XP et Vista, il ya peu de différences entre les deux extensions). N'utilisez pas *Wordpad* ou *Word* à moins que vous ne fassiez très attention à sauvegarder les fichiers au format texte pur. Les commandes elles-mêmes sont souvent très simples et il n'y a pas besoin d'apprendre un langage de programmation. Ceux qui le veulent peuvent explorer les subtilités qui sont disponibles avec les [branchements](#) et les [boucles](#), mais je vais limiter mes explications ici à quelques applications simples pour des tâches de tous les jours. L'accent sera mis sur le gain de temps et d'effort pour certaines choses comme le ménage et la gestion de fichiers simple.

L'exécution d'un fichier batch est très simple, il suffit de double-cliquer dessus. Les fichiers batch peuvent aussi être lancés dans une invite de commande ou dans l'invite **Démarrer-Exécuter**. Dans ce cas, vous devez utiliser le chemin complet à moins que le fichier ne se trouve dans l'un des chemins du [PATH](#)

XXIV-B - Construire un fichier Batch

Pour la suite de ce chapitre, je vais supposer que vous avez lu la page d'[introduction](#) et la page sur les [commandes](#).

La première ligne d'un fichier batch est souvent la commande

```
@echo off
```

Par défaut, un fichier batch va afficher les commandes au fur et à mesure de leur exécution. Le but de cette première commande est de désactiver cet affichage. La commande « echo off » désactive l'affichage pour le script entier, sauf pour la commande « echo off » elle-même. Le signe « @ » en début de ligne masque également la commande « echo off ». Cette nuance n'est pas vraiment importante pour tous dans notre contexte, mais je la mentionne, car elle est souvent rencontrée dans les scripts. Les scripts que je vais décrire sont très courts et l'absence de cette ligne ne fera pas une grande différence. Toutefois, pour une question de bonne pratique, nous allons l'entrer dans nos scripts.

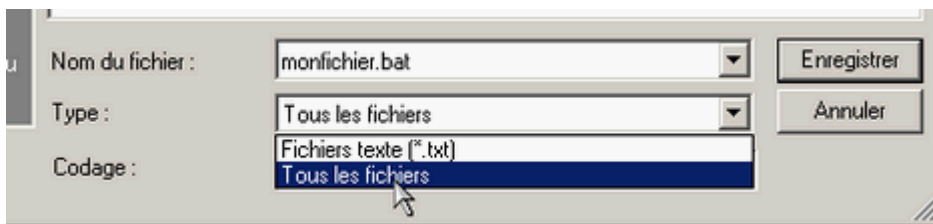
Notre premier exemple de fichier batch va lister tous les fichiers d'un répertoire et placer la liste dans un nouveau fichier texte. Nous allons utiliser la commande « dir » décrite sur [une autre page](#). Ouvrez le bloc note et entrez la ligne « @echo off » (sans les guillemets). Sur une autre ligne, écrivez ensuite

```
dir "C:\Program Files" > C:\list_of_program_files.txt
```

(je suppose ici que votre dossier Program Files se trouve sur le disque C:). Cela va donc nous donner un fichier de deux lignes

```
@echo off
dir "C:\Program Files" > C:\list_of_program_files.txt
```

Enregistrer maintenant ce fichier sous « listeprogrammes.bat » (sans guillemets) dans un emplacement de votre choix. Vérifiez que Notepad enregistre sous le type « Tous les fichiers » et non pas « fichier texte ». Voyez l'image ci-dessous.



Trois points importants sont illustrés dans cet exemple. Remarquez que l'on a utilisé les chemins complets y compris la lettre du lecteur. Remarquez également les guillemets autour de « C:\Program Files ». Les chemins doivent être entourés de guillemets s'il y a un espace dans un nom de fichier ou de dossier. Enfin, notez que le [symbole de redirection](#) « > » est utilisé pour envoyer le résultat dans un fichier plutôt que vers l'écran.

Pour utiliser ce fichier, vous devez simplement double-cliquer dessus. Un fichier appelé *C:\list_of_program_files.txt* sera alors créé.

XXIV-B-1 - Une version plus générale avec des arguments

Le fichier dont nous avons parlé précédemment est limité à la liste d'un répertoire en particulier et place la liste dans un fichier particulier. Cependant, il est facile de créer un script capable de lister le dossier de votre choix et de placer la liste dans le fichier que vous souhaitez. Les fichiers batch peuvent utiliser des arguments ou des données saisies par l'utilisateur. Le processus fait appel à des paramètres de la forme %1, %2. Ils sont remplacés dans le script par les données entrées par l'utilisateur. Dans ce cas, le fichier ne peut pas être lancé simplement en cliquant, mais doit être exécuté à partir de l'invite de commandes. Le nouveau fichier batch sera alors :

```
@echo off
dir %1 > %2
```

Saisissez-le dans le bloc-notes et enregistrez-le sous « makelist.bat ». Pour exécuter le script, ouvrez une invite de commandes et tapez

```
{chemin}makelist unDossier quelquepart\liste.txt
```

où *unDossier* est le dossier (avec un chemin complet) dont vous voulez obtenir la liste dans *quelquepart\liste.txt*. Vous avez maintenant un petit programme qui liste du répertoire que vous choisissez. Si vous voulez aussi la liste des sous-répertoires, utilisez la commande

```
dir /s %1 > %2
```

Si vous voulez une liste qui n'inclut que certains types de fichiers, les fichiers MP3 par exemple, utilisez :

```
dir %1\*.mp3 > %2
```

La ligne précédente illustre l'utilisation du joker « * ». La possibilité d'utiliser des jokers augmente grandement la puissance des fichiers batch.

Vous pouvez vous simplifier la vie en mettant tous vos scripts batch dans un dossier qui est dans le **PATH**

XXIV-B-2 - L'instruction Rem

Les fichiers batch contiennent très souvent des lignes qui commencent par « Rem ». Il s'agit d'un moyen permettant d'insérer des commentaires et de la documentation. Lors de l'exécution, l'ordinateur ignore tout ce qui se trouve sur une ligne commençant par « Rem ». Utiliser des commentaires dans les fichiers batch complexes peut être une bonne idée. Notez toutefois que l'interpréteur de commande lit tout de même les lignes « Rem » donc si vous en utilisez beaucoup trop, cela peut ralentir l'exécution du script.

XXIV-B-3 - Plus d'exemples

Après avoir lu cette [autre page](#), il est facile de créer des fichiers batch pour effectuer de la maintenance. Pour créer un script de sauvegarde simple, utilisez **xcopy**. Le code pourrait être :

```
xcopy %1 %2 /d /s
```

Cette commande va mettre à jour tous les fichiers du répertoire source %1 et de ses sous-répertoires en les copiant dans le dossier de sauvegarde %2. En pratique, un script de sauvegarde utile aura besoin de plus de paramètres décrits dans [le chapitre Xcopy](#).

De même avec les [explications précédentes](#) sur la commande « del », voici un script permettant de supprimer tous les fichiers portant l'extension TMP :

```
del %1\*.tmp
```

XXIV-B-4 - Demander une saisie à l'utilisateur

Vous pouvez aussi interagir avec l'utilisateur et lui demander de saisir des données. L'ancien DOS avait la commande « Choice » pour une interaction très limitée, mais celle-ci a été remplacée dans Windows XP/Vista par la commande « set /p » qui est plus polyvalente. La syntaxe est :

```
set /p variable= [chaine_de_caracteres]
```

« Variable » correspond au nom de la variable qui contiendra les données saisies par l'utilisateur. « chaine_de_caracteres » correspond au message qui s'affichera à l'utilisateur. Vous pouvez omettre « chaine de caracteres » Voici un exemple qui demande son nom à l'utilisateur :

```
set /p nom= Quel est votre nom ?
```

Cela va créer la variable %nom% dont la valeur contiendra la saisie de l'utilisateur. Notez que l'utilisateur doit taper « Entrée » à la fin de sa saisie

(La commande « Choice » est de nouveau disponible dans Vista sous une forme plus puissante).

XXIV-C - Pour en savoir plus

Les exemples présentés ici sont simples et cette page ne prétend pas tout expliquer sur les fichiers batch. Le but de ce chapitre est de montrer combien ces scripts sont simples et d'encourager le lecteur à vouloir en savoir plus à leur sujet. Des fichiers batch plus puissants peuvent même être écrits en ajoutant simplement les prises de décision et des méthodes pour effectuer plusieurs fois les mêmes choses. Les branchements avec « if » et « goto » sont [détaillés ici](#) ; l'utilisation de « For » pour [effectuer des tâches répétitives](#) est expliquée sur une autre page.

Les fichiers batch sont expliqués dans de nombreux livres sur Windows, sur de nombreux sites web et sur [ce site Microsoft](#). Même si vous ne voulez pas en écrire, il en existe quelques-uns prêts à être utilisés. Cet [autre site](#) liste de nombreuses sources.

XXV - Des fichiers batch plus puissants : Partie 1 - Branchements et boucles



Ce chapitre est une traduction de la page  <http://commandwindows.com/batchfiles-branching.htm> dont l'auteur original est Victor Laurie.

Ce chapitre explique les commandes « if...else » et « goto ».

Bien que de nombreuses tâches utiles puissent être réalisées avec de simples fichiers batch contenant uniquement quelques lignes, toute la puissance qui est disponible ne peut être exploitée sans l'utilisation des méthodes de branchement, d'itération et de boucle. Ces méthodes font partie des outils utilisés par les programmeurs pour créer des scripts très sophistiqués. Toutefois, les concepts sont en fait assez faciles à comprendre et sont accessibles à ceux qui n'ont pas de connaissances en programmation. Quelques lignes supplémentaires dans un batch en utilisant cet outil peuvent augmenter significativement la puissance et la polyvalence du script. Dans ce chapitre, je vais expliquer les branchements. **Dans la deuxième partie**, je présenterai les méthodes itératives.

XXV-A - Branchement conditionnel avec l'instruction « if »

Les fichiers batch peuvent prendre des décisions et choisir des actions en fonction de certaines conditions. Pour cela, on utilise des instructions commençant par « if ». La signification de base d'une instruction « if » (si) est la suivante :

```
IF une condition est vraie ALORS faire une action (SINON, faire une autre action)
```

La seconde partie de l'instruction (entre parenthèses) est optionnelle. Sinon, si la première condition n'est pas remplie, le système va exécuter la ligne suivante dans le fichier batch. La syntaxe est :

```
if (condition) (commande1) else (commande2)
```

La partie « Else » est facultative. Vous pouvez aussi utiliser la forme « if not » pour tester si une condition est fausse. Notez que « if » teste vrai et faux dans le sens booléen.

XXV-A-1 - Instruction « if exist »

Il existe une instruction spéciale « if exist » qui peut être utilisée pour tester l'existence d'un fichier, suivie d'une commande. Voici un exemple :

```
if exist unFichier.ext del unFichier.ext
```

Vous pouvez également utiliser un test à la forme négative :

```
if not exist unFichier.ext echo Pas de fichier
```

XXV-A-2 - Instruction « if defined »

Un autre cas spécial est l'instruction « if defined », celle-ci est utilisée pour tester l'existence d'une variable. Par exemple :

```
if defined uneVariable uneCommande
```

Vous pouvez aussi utiliser la forme négative : « if not defined ».

XXV-A-3 - Instruction « if errorlevel »

L'instruction « if errorlevel » est encore un autre cas spécial, qui est utilisé pour tester le code de sortie de la dernière commande qui a été exécutée. Plusieurs commandes renvoient un code de sortie qui indique l'état de la commande. En général, les commandes renvoient 0 si elles se sont exécutées correctement et 1 si la commande a échoué. Certaines commandes peuvent renvoyer d'autres valeurs de code de retour. Par exemple, la commande **Xcopy** peut renvoyer cinq codes de sortie différents. Ces codes de sortie sont enregistrés dans une variable spéciale appelée **errorlevel**. Voici un exemple d'utilisation :

```
if errorlevel n uneCommande
```

où « n » est l'un des codes de sortie (entier) possibles. Notez que la comparaison est effectuée en vérifiant si **errorlevel** est supérieur ou égal à *n*. Si vous utilisez « not », la comparaison vérifie si **errorlevel** est inférieur à *n*

Tableau I. Comparaison des opérateurs dans l'instruction « if »

Opérateur	Signification
EQU	égal à
NEQ	différent de
LSS	Plus petit que (Less Than)
LEQ	Inférieur ou égal à (Less or Equal)
GTR	Plus grand que (Greater Than).
GEQ	Supérieur ou égal à (Greater or Equal)

XXV-A-4 - Opérateurs de comparaison

Dans certains cas, la condition à remplir est une comparaison de chaînes de caractères. Par exemple :

```
if chain1 == chaine2
```

Remarquez que le signe « égal » est écrit en double. La condition est vraie si les deux chaînes de caractères sont strictement identiques, y compris la casse. Pour rendre la comparaison insensible à la casse, utilisez le commutateur « /i ». Pour des comparaisons plus générales, utilisez les opérateurs du tableau I. (Les opérateurs sont donnés en majuscules dans le tableau, mais ils sont insensibles à la casse). Les comparaisons numériques ne fonctionnent qu'avec des chaînes ne contenant que des chiffres. Sinon, la comparaison est faite alphabétiquement. Par exemple « a » est inférieur à « b ». Pour ne pas tenir compte de la casse, utiliser le commutateur « /i ». Voici un exemple de commande :

```
if /i chain1 gtr chaine2 uneCommande
```

Quand vous comparez des variables chaînes de caractères, il est préférable d'entourer le nom de la variable avec des guillemets. Vous pouvez par exemple utiliser :

```
if "%1" == uneChaine uneCommande
```

XXV-B - La commande « goto »

Généralement, l'exécution d'un fichier batch se déroule en exécutant à tour de rôle les commandes ligne par ligne. Cependant, on souhaite souvent exécuter une section particulière du fichier batch et sauter d'autres parties. La possibilité de sauter à une section particulière est fournie par la commande appelée « goto » (en un seul mot). La section de destination possède comme label un nom précédé d'un symbole deux-points (:). Ainsi, le script ressemble à cela :

```
...  
goto :label  
...{quelques commandes}...  
:label  
...{d'autres commandes}...
```

À l'exécution les « quelques commandes » seront sautées et les « autres commandes » seront exécutées. Le label peut être situé sur une ligne à n'importe quel endroit du script, y compris avant la commande « goto ».

Les commandes « goto » sont souvent utilisées avec les instructions « if ». Vous pouvez par exemple utiliser une commande de ce type :

```
if (condition) goto :label
```

XXV-B-1 - Le label "End of File (:eof) est utilisé pour sortir d'un script

Il est parfois utile de terminer un script si une certaine condition est remplie (ou non). L'une des façons possibles est d'utiliser le label spécial :**eof** dans une commande **goto**. Ce label n'est pas présent dans le fichier. Windows XP et les versions suivantes reconnaissent :**eof** sans que ce label soit placé explicitement à la fin du fichier batch. Ainsi, si vous avez besoin de tester une condition particulière devant entraîner l'arrêt du script, vous pouvez écrire :

```
if (condition) goto :eof
```

Notez que cela termine le script, mais que cela ne ferme pas forcément l'invite de commande.

XXV-C - Boucles avec « if » et « goto »

Une ancienne méthode pour effectuer des tâches répétitives était d'utiliser un compteur, des instructions « if » et la commande « goto ». Le compteur détermine combien de fois la tâche doit être répétée, l'instruction « if » détermine le moment où le nombre de répétitions désiré est atteint, et la commande « goto » permet d'exécuter l'action appropriée : soit la tâche à répéter soit quitter. Généralement, il est préférable d'utiliser la méthode plus élégante proposée par la puissante commande « for...in...do ». Cette commande est décrite dans [le chapitre suivant](#). Toutefois, par souci d'exhaustivité et pour illustrer ce que je vous ai expliqué, je vais donner un exemple qui utilise la méthode avec un compteur.

Le script d'exemple ci-dessous crée des nombres entre 1 et 99 et les enregistre dans un fichier. Ce script utilise la **commande « set »** pour créer une variable qui sert de compteur pour savoir combien de fois on a bouclé.

```
@echo off  
set /a counter=0  
  
:numbers  
set /a counter=%counter%+1  
if %counter% ==100 (goto :eof) else (echo %counter% >> E:\count.txt)  
goto :numbers
```


(Les consignes de meilleure programmation indiquent que la variable **%counter%** devrait être localisée ou détruite à la fin, mais par simplicité, j'ai omis les lignes supplémentaires pour le faire. Tel que c'est écrit, la variable d'environnement va persister jusqu'à ce que l'invite de commande (pas juste le script) soit fermée.

En anticipant un peu, je vous faire remarquer qu'il est possible d'obtenir le résultat du script ci-dessus avec un script de deux lignes utilisant l'instruction « for » décrite [au chapitre suivant](#) :

```
@echo off
for /l %%X in (1,1,99) do (echo %%X >> E:\count.txt)
```

XXVI - Des fichiers batch plus puissants : Partie 2 - Itérations avec « For »



Ce chapitre est une traduction de la page <http://commandwindows.com/batchfiles-iterating.htm> dont l'auteur original est Victor Laurie.

Je vais décrire ici la puissante instruction « for...in...do ».

Les ordinateurs sont très forts pour répéter la même opération encore et encore... La ligne de commande contient une méthode puissante et polyvalente pour effectuer ce type d'opérations. Avec cette méthode, vous pouvez automatiser de nombreuses tâches consommatrices de temps. L'instruction de base est de la forme :

```
for
{chaque élément}
in
{un ensemble d'éléments}
do
{commande}
```

Une variable d'une seule lettre est utilisée pour représenter chaque élément lorsque la commande parcourt l'ensemble. Notez que, contrairement à la plupart des autres commandes Windows, les variables sont sensibles à la casse. Ainsi, « a » et « A » sont deux variables différentes. La variable n'a aucune signification à l'extérieur de l'instruction « For ». Je vais utiliser la variable X dans la suite de ce chapitre, mais vous pouvez utiliser n'importe quelle autre lettre. (En principe, certains caractères non alphanumériques peuvent aussi être utilisés, mais il me semble que c'est une mauvaise idée). La variable d'une lettre est précédée par un symbole % lorsqu'elle est utilisée en ligne de commande et doit être précédée de deux caractères % si elle est utilisée dans un fichier batch. Ainsi, l'instruction dans un fichier batch ressemblera à ceci :

```
for %%X in (ensemble) do (commande)
```

Ce qui fait de « For » une commande puissante est la variété d'objets qui peuvent être utilisés comme « ensemble » à parcourir, la disponibilité des caractères jokers et la possibilité de parcourir des fichiers et les sorties de commandes. Un certain nombre de commutateurs sont disponibles pour aider à définir le type d'éléments dans l'ensemble à parcourir. Le tableau I liste ces commutateurs. Ils sont listés en majuscules pour plus de clarté, mais la casse n'est pas importante.

Tableau I. Commutateurs utilisés avec FOR

Commutateur	Fonction
/D	Indique que l'ensemble contient des répertoires.
/R	La commande est exécutée récursivement dans les sous-répertoires du répertoire parent indiqué.
/L	Boucle sur une commande en utilisant des paramètres de début, de fin et de pas indiqués dans l'ensemble.
/F	Parcours des fichiers ou une sortie de commande de plusieurs manières.

Je vais donner quelques exemples pour illustrer l'utilisation de « For » et de ses commutateurs.

XXVI-A - Itération simple dans une liste

L'ensemble d'éléments parcourus peut être listé explicitement. Par exemple, l'ensemble pourrait être une liste de fichiers :

```
for %%X in (fichier1 fichier2 fichier3) do commande
```

(Vous devez faire attention à utiliser les bons chemins quand vous effectuez des opérations de fichiers). Un autre exemple où les éléments sont des chaînes de caractères :

```
for %%X in (eenie meenie miney moe) do (echo %%X)
```

Vous pouvez aussi utiliser les jokers pour désigner un jeu de fichiers. Par exemple :

```
for %%X in (*.jpg) do commande
```

Cette syntaxe va exécuter « commande » sur tous les fichiers « jpg » du répertoire courant. Ce processus peut être enrichi en utilisant plusieurs membres dans l'ensemble. Par exemple pour exécuter une commande sur plus d'un type de fichiers, utilisez :

```
for %%X in (*.jpg *.gif *.png *.bmp) do commande
```

Comme toujours, gardez à l'esprit que la ligne de commande n'aime pas les noms de fichiers avec des espaces à moins que le nom soit correctement entouré de guillemets. Par conséquent, vous voudrez peut-être utiliser « %%X » pour la « commande ».

XXVI-B - Boucler sur une série de valeurs

L'action bien connue de parcourir une série de valeur **qui a été décrite** dans le cadre des instructions « if » et « goto », se fait rapidement avec le commutateur /l (Ce commutateur est la lettre « L », et non pas un « un »). L'instruction est de la forme :

```
for /l %%X in (début, pas, fin) do commande
```

. L'ensemble consiste en entiers définissant la valeur initiale de X, la valeur d'incrément (ou décrétement) de X, et la valeur finale de X quand le processus se terminera. Sur la [page précédente](#), j'ai donné un exemple de fichier batch qui listait tous les nombres de 1 à 99. Si nous utilisons une instruction « For », cette tâche peut être accomplie avec une ligne :

```
for /l %%X in (1,1,99) do (echo %%X >> E:\nombres.txt)
```

. Les nombres dans l'ensemble signifient que la valeur initiale de X est 1, que la variable X est incrémentée de 1 à chaque itération et que la valeur finale de X est 99.

XXVI-C - Utiliser les répertoires

Si vous souhaitez utiliser les répertoires dans la variable set, utilisez le commutateur /d. La syntaxe de la commande est la suivante :

```
for /d %%X in (ensembleDeRepertoires) do commande
```

. Voici un exemple qui affiche tous les répertoires (mais pas les sous-répertoires) présents sur le disque « C : » :

```
for /d %%X in (C:\*) do echo %%X
```

XXVI-D - Parcourir récursivement les sous-répertoires

Si vous voulez appliquer une commande au répertoire parent ainsi qu'à ses sous-répertoires, utilisez le commutateur /r. La commande est alors de la forme :

```
for /r [repertoireParent] %%X in (ensemble) do commande
```

Cela permet souvent d'éviter le pénible problème de savoir quel est le répertoire de travail pour la commande shell. Par exemple, l'instruction :

```
for /r C:\photos %%X in (*.jpg) do (echo %%X >> E:\listeJPG.txt)
```

va lister tous les fichiers jpg dans le répertoire c:\photos et ses sous-répertoires. Bien sûr, une commande « dir » permet d'obtenir la même chose, mais c'est simplement pour illustrer cette commande particulière.

XXVI-E - Parcourir des fichiers, chaînes et sorties de commandes

Nous arrivons maintenant à un commutateur vraiment puissant qui n'avait même pas été imaginé du temps de DOS. Le commutateur /f nous amène dans un territoire avancé, donc je ne peux indiquer que les principaux aspects de ses utilisations. Les choses peuvent être assez complexes donc ceux qui sont intéressés peuvent consulter des livres de programmation ou la **documentation Microsoft**. Cependant, voici un aperçu de ce que l'on peut faire.

Cette version de la commande « for » vous permet d'examiner et de parcourir le texte de fichiers, les chaînes de caractères et les sorties de commandes. La syntaxe est :

```
for /f [options] %%X in (source) do commande
```

Les « options » représentent le critère de recherche dans le texte, et la « source » indique où le texte doit être cherché. L'une des utilisations les plus intéressantes est l'analyse de la sortie d'une ou plusieurs commandes et la prise de décision basée sur ce qu'ont renvoyé la ou les commandes.

Pour en savoir plus, vous pouvez consulter l'aide de la commande *for* en utilisant la commande suivante :

```
help for
```

XXVII - Variables dans l'invite de commandes Windows



Ce chapitre est une traduction de la page <http://commandwindows.com/variables.htm> dont l'auteur original est Victor Laurie.

Dans ce chapitre, je vais décrire la déclaration et l'utilisation de variables avec la commande « set ».

Les variables ont une place importante dans de nombreux langages de script, mais elles jouent un rôle moindre dans la ligne de commande Windows. De nombreuses commandes sont prédéfinies et la portée des variables est assez limitée. Néanmoins, il existe d'importantes utilisations de la ligne de commande où les variables doivent être employées et dans ce chapitre, je vais décrire comment la ligne de commande utilise des variables.

XXVII-A - Comment les variables sont définies avec la commande « set »

On peut dire qu'il existe deux catégories de variables pour la ligne de commande. Certains pourraient utiliser le terme « variable » pour les arguments %1, %2, ..%9, qui sont utilisés pour représenter la saisie de l'utilisateur dans les fichiers batch. ([Consultez la description sur cette page](#)). Cependant, le terme « variable » est normalement réservé dans l'invite de commande pour les éléments déclarés comme des **variables d'environnement** avec la commande « set ». Notez qu'il s'agit d'une méthode primitive pour définir des variables. Par exemple, il n'y a pas de types. Les variables d'environnement, y compris les nombres, sont stockées en tant que chaînes de caractères et les opérations qui les utilisent doivent prendre ce fait en compte. Les variables sont déclarées et une valeur leur est assignée en une seule instruction utilisant « set ». La syntaxe est :

```
set une_variable = une_valeur
```

Les noms de variables ne sont pas sensibles à la casse et peuvent être constitués des caractères alphanumériques usuels et autres caractères courants. Certains caractères sont réservés et doivent être « échappés ». Bref, il vaut mieux les éviter. Ces symboles sont listés dans [le tableau II sur cette page](#). Aussi, étant donné que ce sont des variables d'environnement, leurs noms doivent être entourés de symboles % (pour cent) lorsqu'ils sont utilisés dans les références et expressions, *par exemple* :

```
%une_variable%
```

Le signe pour cent n'est pas utilisé dans la partie gauche de l'instruction « set » qui déclare une variable.

XXVII-A-1 - Localiser les variables

La déclaration d'une variable dure tout le temps pendant lequel la fenêtre de commande courante est ouverte. Si vous utilisez un fichier batch qui ne ferme pas la fenêtre de commande à la fin du traitement, toutes les variables déclarées par le batch restent définies. Si vous souhaitez limiter la portée d'une variable à un ensemble d'instructions particulier, utilisez les commandes « setlocal » et « endlocal ». Ainsi, pour limiter la déclaration d'une variable à un bloc de code particulier, utilisez :

```
....
setlocal
set une_variable = une_valeur
...{des instructions}...
endlocal
....
```

XXVII-A-2 - Variables saisies par l'utilisateur

La commande « set » peut également accepter une saisie de l'utilisateur comme valeur pour une variable. Pour cela, on utilise le commutateur « /p ». Un fichier batch attendra que l'utilisateur entre une valeur après l'instruction

```
set /p nouvelle_variable=
```

Quand l'utilisateur aura saisi une valeur, le script continuera. Vous pouvez aussi définir un message d'invite pour l'utilisateur. Par exemple :

```
set /p nouvelle_variable="Entrez une valeur "
```

Remarquez l'espace à la fin du message d'invite. Si vous ne le faites pas, le message d'invite et l'entrée de l'utilisateur se retrouveront côte à côte sur l'écran. Cela marche, mais l'apparence n'est pas fameuse. L'utilisateur peut être tenté de taper une espace, cela ajoutera alors un espace à la valeur saisie.

XXVII-B - Opérations arithmétiques

La ligne de commande n'est pas prévue pour gérer les fonctions mathématiques, mais il est possible d'exécuter des opérations très simples d'entiers avec les variables. Un commutateur « /a » a été ajouté à la commande « set » pour permettre quelques fonctions basiques. Cela est principalement utilisé pour effectuer des additions ou des soustractions. Par exemple, il est possible d'incrémenter ou de décrémenter un compteur dans une boucle. En principe, il est également possible d'effectuer des multiplications et des divisions, mais seuls quelques nombres peuvent être gérés donc l'utilisation en pratique est limitée. Bien que les variables soient stockées comme des chaînes, l'interpréteur de commande reconnaît les chaînes qui ne contiennent que des entiers, et leur permet d'être utilisées dans des expressions arithmétiques. La syntaxe est :

```
set /a une_variable={expression arithmétique}
```

Les quatre opérations arithmétiques sont indiquées dans le tableau I. J'ai omis l'opération « modulo » qui utilise le signe % d'une autre manière. À mon avis, cela ne fait qu'ajouter de la difficulté à une syntaxe déjà compliquée. (L'utilisation du % de plusieurs manières ne peut amener que de la confusion).

Tableau I. Opérateurs arithmétiques

Symbole	Opération
+	Addition
-	Soustraction
*	Multiplication
/	Division

Voici l'exemple d'une variable %compteur% incrémentée :

```
set /a compteur=%compteur%+1
```

Vous pouvez également écrire :

```
set /a compteur+=1
```

XXVII-C - Instruction de comparaison de variables dans les fichiers batch

Les variables sont souvent utilisées pour des comparaisons dans les instructions conditionnelles (if...) des fichiers batch. Certains des opérateurs de comparaison utilisés sont donnés dans le tableau I de [la page sur l'instruction](#)

« if ». En raison de la façon dont la ligne de commande gère les variables, il est nécessaire d'être prudent quand on compare des variables. Pour les chaînes, le moyen le plus sûr est d'entourer les variables de guillemets. par exemple :

```
if "%variable1%" == "%variable2%" une_commande
```

XXVIII - Doskey dans l'invite de commandes Windows XP



Ce chapitre est une traduction de la page <http://commandwindows.com/doskey.htm> dont l'auteur original est Victor Laurie.

La console de commande Windows XP contient un accessoire nommé Doskey comme l'ancien utilitaire en ligne de commande. Ces fonctions sont décrites dans ce chapitre.

Comme son nom l'indique, Doskey faisait partie de l'ancienne invite de commande DOS. Elle fournissait l'édition en ligne de commande, des fonctions d'historique pour réappeler des commandes, et un moyen pour écrire des macros. Un outil de même nom est toujours présent dans les invites de commande 32-bits fournis par `cmd.exe`, mais ses fonctions ont été grandement remaniées. Par exemple, si les extensions sont activées (le réglage par défaut), un **tampon mémorise les lignes de commande saisies** permet de ré-appeler les 50 dernières commandes tapées sans avoir besoin de Doskey. Vous pouvez utiliser les flèches Haut et Bas pour naviguer parmi les commandes entrées précédemment. Sauf si vous travaillez pendant de longues périodes dans la console de commande, je ne vois pas l'avantage d'utiliser des macros plutôt que des fichiers batch. J'écris ce chapitre par souci de complétude, mais je n'ai pas utilisé personnellement Doskey depuis des années.

XXVIII-A - Syntaxe et fonctions

La figure ci-dessous résume la syntaxe de Doskey et de ses options et fonctions

```
C:\>doskey /?
Édite des lignes de commande, mémorise des commandes Windows XP
et crée des macros.
DOSKEY [/REINSTALL] [/LISTSIZE=taille] [/MACROS[:ALL ; :nom_d'exel]]
[/HISTORY] [/INSERT ; /OVERSTRIKE] [/EXENAME=nom_d'exel]
[/MACROFILE=nom_de_fichier] [nom_de_macro=[texte]]

/REINSTALL          Installe une nouvelle copie de Doskey.
/LISTSIZE=taille    Définit la taille du tampon d'historique des commandes
/MACROS             Affiche toutes les macros de Doskey.
/MACROS:ALL         Affiche toutes les macros de Doskey de tous les
                   exécutable qui en ont.
/MACROS:nom_d'exe   Affiche toutes les macros de Doskey d'un exécutable.
/HISTORY            Affiche toutes les commandes stockées dans la mémoire.
/INSERT            Insertion dans l'ancien texte du nouveau texte entré.
/OVERSTRIKE         Spécifie que le nouveau texte remplace l'ancien.
/EXENAME=nom_d'exe Spécifie l'exécutable.
/MACROFILE=nom_de_fic. Spécifie un fichier de macros à installer.
nom_de_macro        Spécifie un nom pour une macro que vous créez.
texte               Spécifie des commandes que vous voulez enregistrer.

Flèches HAUT/BAS rappellent les commandes ; ÉCHAP efface la ligne ; F7 affiche
l'historique ; ALT+F7 efface l'historique ; F8 recherche dans l'historique ;
F9 choisit une commande par son nombre ; ALT+F10 efface les déf. de macros.

Les codes spéciaux suivants s'utilisent pour définir les macros Doskey :
$! Séparateur de commande. Autorise plusieurs commandes dans une macro.
$1-$9 Paramètres de batch. Comme %1-%9 dans les programmes de commandes.
$* Équivaut à tout ce qui suit le nom de macro sur la ligne de commande.
```

XXVIII-B - Plus d'informations

Une description plus complète de Doskey est disponible dans [cette référence Microsoft](#).

XXIX - PowerShell dans Windows XP



Ce chapitre est une traduction de la page <http://commandwindows.com/powershell.htm> dont l'auteur original est Victor Laurie.

Microsoft a introduit une interface de ligne de commande complètement nouvelle appelée « PowerShell ». Ce chapitre décrit quelques nouvelles fonctionnalités.

XXIX-A - Introduction à PowerShell

Microsoft souhaitait à l'origine développer une nouvelle interface en ligne de commande (nom de code « Monad ») intégrée à Vista. Cependant, il fut ensuite décidé de faire de la fonctionnalité (renommée « PowerShell ») une application indépendante qui est **maintenant disponible** pour Windows XP. (La version pour Vista est également publiée et est **disponible sur ce site**. D'ailleurs, Vista possède encore l'interpréteur de commande `cmd.exe`, tout comme Windows XP). Le nouveau shell est la réponse de Microsoft au scripting shell Unix. Il est tout à fait différent de la précédente interface de ligne de commande et est considérablement plus puissant. Il utilise des techniques plus sophistiquées et **les objets** et il nécessite le Framework .NET 2.0. Il dispose de nouvelles fonctions pour les systèmes et l'administration réseau et est destiné aux professionnels IT. Parce que le but de cet article est de présenter la ligne de commande aux utilisateurs de PC, PowerShell dépasse un peu le cadre de cet article (et mon expérience personnelle). Néanmoins, je crois que l'utilisateur doit être conscient du potentiel de PowerShell et les plus expérimentés peuvent souhaiter l'explorer plus en détail. Je vais essayer de décrire brièvement ce qu'est PowerShell.

XXIX-B - Les fonctionnalités de PowerShell

Dans l'invite de commande précédemment décrite dans cet article, les commandes consistent en une série de commandes internes sous forme de chaînes de caractères qui sont interprétées et exécutées par l'interpréteur de commandes ou de commandes invoquant des fichiers exécutables séparés. PowerShell a une nouvelle approche qui permet l'utilisation de ce que Microsoft appelle des « cmdlets ». Voici la description de Microsoft : « Une cmdlet (prononcez »command-let«) est une commande à fonction unique qui manipule des objets dans Windows PowerShell. Vous pouvez reconnaître les cmdlets par leur format de nom -- un verbe et un nom séparés par un tiret (-), comme `Get-Help`, `Get-Process`, et `Start-Service`. »

Bien que chaque cmdlet ait une fonction unique, des groupes de cmdlets peuvent être reliés ensemble pour mener à bien une tâche complexe. Le résultat de plusieurs cmdlets pourra également être utilisé comme entrée d'autres cmdlets (par un « pipe ») sans traitement supplémentaire. Ces possibilités nouvelles représentent une avancée significative par rapport à la console de lignes de commandes.

PowerShell continue à reconnaître les commandes provenant de l'ancienne console de commande bien que, dans plusieurs cas, la commande soit un alias d'une cmdlet PowerShell.

XXIX-B-1 - Liste de cmdlets

Pour l'instant, PowerShell est fourni avec 129 cmdlets. Comme les cmdlets sont faciles à écrire, on peut s'attendre à ce que ce nombre augmente. Le tableau I indique celles présentes à la date d'écriture de ce document.

Tableau I. Liste des cmdlets PowerShell

Add-Content	Get-Date	Move-ItemProperty	Set-
Add-History	Get-EventLog	New-Alias	AuthenticodeSignature
Add-Member	Get-ExecutionPolicy	New-Item	Set-Content
Add-PSSnapin	Get-Help	New-ItemProperty	Set-Date
Clear-Content	Get-History	New-Object	Set-ExecutionPolicy

Clear-Item	Get-Host	New-PSDrive	Set-Item
Clear-ItemProperty	Get-Item	New-Service	Set-ItemProperty
Clear-Variable	Get-ItemProperty	New-TimeSpan	Set-Location
Compare-Object	Get-Location	New-Variable	Set-PSDebug
ConvertFrom-SecureString	Get-Member	Out-Default	Set-Service
Convert-Path	Get-PfxCertificate	Out-File	Set-TraceSource
ConvertTo-Html	Get-Process	Out-Host	Set-Variable
ConvertTo-SecureString	Get-PSDrive	Out-Null	Sort-Object
Copy-Item	Get-PSProvider	Out-Printer	Split-Path
Copy-ItemProperty	Get-PSSnapin	Out-String	Start-Service
Export-Alias	Get-Service	Pop-Location	Start-Sleep
Export-Clixml	Get-TraceSource	Push-Location	Start-Transcript
Export-Console	Get-UICulture	Read-Host	Stop-Process
Export-Csv	Get-Unique	Remove-Item	Stop-Service
ForEach-Object	Get-Variable	Remove-ItemProperty	Stop-Transcript
Format-Custom	Get-WmiObject	Remove-PSDrive	Suspend-Service
Format-List	Group-Object	Remove-PSSnapin	Tee-Object
Format-Table	Import-Alias	Remove-Variable	Test-Path
Format-Wide	Import-Clixml	Rename-Item	Trace-Command
Get-Acl	Import-Csv	Rename-ItemProperty	Update-FormatData
Get-Alias	Invoke-Expression	Resolve-Path	Update-TypeData
Get-	Invoke-History	Restart-Service	Where-Object
AuthenticateSignature	Invoke-Item	Resume-Service	Write-Debug
Get-ChildItem	Join-Path	Select-Object	Write-Error
Get-Command	Measure-Command	Select-String	Write-Host
Get-Content	Measure-Object	Set-Acl	Write-Output
Get-Credential	Move-Item	Set-Alias	Write-Progress
Get-Culture			Write-Verbose
			Write-Warning

XXIX-B-2 - Syntaxe des cmdlets

Il existe un certain nombre de paramètres possibles pour les cmdlets, mais une description détaillée de leur syntaxe sortirait du cadre de cet article. Je vais essayer de vous montrer l'éventail des possibilités offertes en vous décrivant une cmdlet utile qui effectue la fonction de copie. Elle n'est pas limitée à la copie de fichiers et dossiers, mais peut également copier des clés et entrées du Registre. Cette cmdlet, en fait, incorpore les fonctions de plusieurs anciennes commandes avec une plus grande souplesse. Tout d'abord, voyons un exemple où un dossier et tout son contenu doivent être copiés :

```
Copy-Item C:\Logfiles -destination D:\Backup -recurse
```

Cette cmdlet copie tous les fichiers et sous-dossiers du dossier C:\Logfiles dans le dossier D:\Backup. Le paramètre « -recurse » est utilisé quand les sous-dossiers doivent être copiés.

Voyons maintenant tous les paramètres :

```
Copy-Item [-path] (string[]) [[-destination] (string)] [-container] [-recurse] [-force] [-include (string[])] [-exclude (string[])] [-filter (string)] [-passThru] [-credential (PSCredential)] [-whatIf] [-confirm] [ (CommonParameters) ]
```

Naturellement, l'ensemble complet des paramètres varie d'une cmdlet à une autre, mais l'intrigant paramètre « -whatIf » est commun à toutes. Cette option décrit ce qui devrait se passer si vous exécutiez la commande, mais sans l'exécuter réellement. Cela vous permet de voir en toute sécurité ce qui se passerait si vous utilisiez la commande. Pour un tableau décrivant les différents paramètres indiqués ci-dessus, [cliquez ici](#).

XXIX-C - Scripting PowerShell

PowerShell est aussi la base d'un langage de script. Un aperçu des opérateurs et fonctions disponibles est accessible dans [cette référence MSDN](#). Ce langage est prévu pour rendre les tâches administratives plus faciles et semble destiné à supplanter VBScript dans le futur. L'extension pour les scripts PowerShell est *.PS1*. Plusieurs fonctionnalités de sécurité sont intégrées au moteur de script et le réglage par défaut est d'empêcher l'exécution des scripts. La permission d'exécuter les scripts est contrôlée par une fonctionnalité appelée « Execution Policy ». Vous pouvez obtenir des informations sur cette fonctionnalité avec la commande PowerShell

```
Get-Help about_signing
```

Vous trouverez plus d'informations sur le scripting PowerShell sur [ce site Microsoft](#)

XXIX-D - Plus d'informations

Dans cette courte description de PowerShell, nous n'avons fait qu'en effleurer la surface. Pour ceux qui souhaitent explorer plus profondément le sujet, voici une liste de références :

- [Page principale Microsoft sur PowerShell](#)
- [Article Wikipedia](#)
- [Wiki PowerShell](#)

XXX - Astuces pour l'utilisation de l'invite de commande Windows



Ce chapitre est une traduction de la page <http://commandwindows.com/tipsandtricks.htm> dont l'auteur original est Victor Laurie.

Ce chapitre va vous présenter quelques trucs et astuces utiles dans l'utilisation de la console de commandes Windows.

Il existe certaines astuces dont les livres sur l'invite de commande ne parlent pas. Ou, s'ils le font, la description est enterrée dans un paragraphe discret. Les utilisateurs expérimentés de la ligne de commande connaissent ces astuces. Cependant, la plupart des utilisateurs lambda ne les connaissent pas et je vais donc donner ici quelques « trucs » utiles pour eux. D'après ce que je sais, les astuces que je vais donner fonctionnent dans Windows XP et Windows Vista (sauf les cas indiqués). Ces astuces peuvent ne pas fonctionner pour des versions antérieures de Windows. Les astuces spécifiques à Vista sont données [sur une autre page](#).

XXX-A - Comment afficher une ligne vide dans un fichier batch

Parfois, vous souhaitez afficher dans la console une ou deux lignes blanches. Il n'est pas évident de savoir comment faire... Entrer simplement la commande « echo » ne fonctionnera pas parce qu'elle va afficher le statut actuel de echo. L'astuce est d'entrer

```
echo.
```

Notez que « echo » est suivie d'un point sans espace entre echo et le point.

XXX-B - Forcer « echo » à ne pas parcourir les arguments

L'astuce précédente est un cas particulier d'un mode d'utilisation plus général de la commande « echo ». Bien que la commande soit utilisée pour afficher du texte ou des messages, elle peut aussi prendre des arguments tels que « on » ou « off ». Si vous tapez

```
echo off
```

vous n'allez pas afficher la chaîne « off », mais configurer la commande « echo » elle-même. Pour afficher la chaîne « off », vous devez utiliser

```
echo. off
```

En d'autres termes, placer un point à la fin de « echo » force la commande à afficher ce qu'il y a sur la suite de la ligne, sans vérifier si la chaîne est un cas particulier.

XXX-C - Vérifier si un fichier existe

Une variante spéciale de l'instruction « if » peut être utilisée pour savoir si un fichier existe déjà. L'instruction à utiliser est :

```
if exist unFichier uneCommande
```

L'instruction peut également tester l'absence d'un fichier :

```
if not exist unFichier uneCommande
```

XXX-D - L'utile périphérique « nul »

Le périphérique invisible appelé « nul » peut être utilisé de plusieurs manières. (il est parfois appelé « trou noir »). Tout ce qui y est envoyé disparaît. Il peut être utilisé dans les instructions quand vous ne voulez pas que leurs sorties soient affichées. Par exemple, la commande

```
uneCommande > nul
```

va exécuter « uneCommande », mais toute la sortie de la commande sera envoyée aux oubliettes. Vous souhaitez parfois, dans un fichier batch, qu'aucun message d'erreur éventuel ne s'affiche. Ceci est obtenu en utilisant

```
uneCommande 2> nul
```

Nul peut également être appliqué à une commande qui fonctionne normalement sur les fichiers pour lui faire utiliser un répertoire à la place. Par exemple, il est parfois utile de tester si un répertoire existe. L'instruction

```
if exist unFichier uneCommande
```

va vérifier l'existence d'un fichier, mais ne fonctionnera pas pour un répertoire. Heureusement, tous les répertoires contiennent le périphérique « nul », ainsi, vous pouvez utiliser le code suivant pour tester si un répertoire existe déjà :

```
if exist unRepertoire\nul
```

XXX-E - Arrêter une commande en cours d'exécution

Parfois, vous démarrez une commande et vous vous apercevez qu'elle vous renvoie de grandes quantités d'information sur l'écran. La plupart du temps, vous pouvez terminer la commande en pressant simultanément les touches « Ctrl » et « C ».

XXX-F - Mettre en pause le défilement de l'écran

Si vous utilisez une commande qui renvoie beaucoup de données en sortie, vous pouvez mettre en pause le défilement afin de pouvoir lire ce qui est écrit à l'écran. Pour cela, utilisez la combinaison de touches « Ctrl+S ». Pour continuer le défilement, appuyez à nouveau sur « Ctrl+S ».

XXX-G - Utiliser le glisser-déplacer

Le fait de devoir taper le chemin complet vers un fichier est une source d'erreur. La plupart des gens ne savent pas qu'il est possible de faire glisser un fichier à partir de l'explorateur Windows vers la console pour que le chemin complet du fichier apparaisse. Cela permet d'économiser du temps de frappe. (Ne fonctionne pas dans Vista)

XXX-H - Remonter au répertoire parent

Les utilisateurs d'Unix connaissent cette syntaxe, mais certains utilisateurs de Windows ne la connaissent pas. Pour remonter au répertoire parent du répertoire actuel, entrez

```
cd ..
```

Vous pouvez répéter cette commande pour remonter les niveaux jusqu'à la racine du disque. Vous pouvez également entrer deux points « .. » dans la fenêtre **Démarrer->Exécuter** cela ouvrira le répertoire parent du répertoire de travail par défaut (il s'agit en général de %USERPROFILE%).

XXX-I - Comment changer le répertoire de travail courant vers un répertoire d'un autre disque

Si vous voulez changer le répertoire de travail d'une fenêtre d'invite de commande vers un dossier situé sur un autre disque, la commande « cd » ne fonctionnera pas. Vous devez d'abord entrer la lettre du lecteur suivie de deux-points (:) puis utiliser la commande « cd » suivie du chemin. Cependant, vous pouvez utiliser le commutateur /d pour changer le répertoire courant :

```
cd /d E:\test
```

Vous pouvez également effectuer le changement avec une seule commande si vous utilisez « pushd » à la place de « cd » :

```
pushd E:\test
```

XXX-J - Attention aux espaces dans les noms des fichiers et des dossiers

L'invite de commande ne reconnaît pas les espaces dans les chemins. Tous les chemins comportant des espaces doivent être entourés de guillemets (""). Ce problème revient souvent dans les scripts lorsque certaines variables d'environnement ou des arguments d'entrée sont utilisés. Par mesure de précaution, les variables qui contiennent des noms de fichiers ou de dossiers doivent être placées entre guillemets.

XXX-K - Traitement spécial des variables pour l'instruction « for » dans les fichiers batch

L'instruction « for » est souvent utile, car elle fournit un moyen puissant d'effectuer des itérations. Cette commande a la particularité de nécessiter l'utilisation de doubles signes pour cent pour les itérations dans les fichiers batch. En d'autres termes, la syntaxe dans un fichier batch est :

```
for %%variable In set Do commande
```

Si une boucle « for » est exécutée directement depuis l'invite de commande, seul un signe pour cent est utilisé. La syntaxe est alors :

```
for %variable In set Do commande
```

XXX-L - Variables sensibles à la casse dans les instructions « for »

Contrairement aux systèmes Unix, Windows n'est généralement pas sensible à la casse, cependant, les variables utilisées dans les instructions « for » sont sensibles à la casse. Donc une variable %A est différente de %a.

XXX-M - Mettre un raccourci vers une ligne de commande dans le Menu Démarrer

Si vous utilisez fréquemment l'invite de commandes, vous pouvez créer un raccourci facilement accessible. Ouvrez **Démarrer-->Tous les programmes-->Accessoires** et faites un clic droit sur « Invite de commande ». Choisissez « Ajouter au menu Démarrer » dans le menu contextuel. Vous pouvez aussi aller dans `\WINDOWS\system32` et effectuer un clic droit sur le fichier de l'interpréteur de commande `cmd.exe` et choisissez « Ajouter au menu Démarrer » dans le menu contextuel.

XXX-N - Créer un raccourci vers une commande

Si vous utilisez fréquemment une commande, vous pouvez créer un raccourci vers celle-ci. L'astuce est d'utiliser le commutateur /k pour que l'invite de commande reste ouverte. La commande pour le raccourci est :

```
cmd /k uneCommande.exe
```

(Vous trouverez plus de détails sur la façon de créer un raccourci [sur cette page](#)).

XXX-O - Ouvrir l'explorateur Windows à partir de la ligne de commande

Pour ouvrir l'explorateur Windows sur le répertoire courant de l'invite de commande, utilisez la commande

```
start .
```

Pour ouvrir le répertoire parent, utilisez la commande

```
start ..
```

XXX-P - Utilisation de la commande « Start »

L'astuce précédente est un exemple d'utilisation de la commande « start » utilisée pour appeler une action, un dossier système ou une URL. Par exemple, vous pouvez taper « cookies » dans « Démarrer->Exécuter » pour ouvrir le répertoire système contenant les cookies (fonctionne dans Windows XP, mais pas dans Vista). Toutefois, dans l'invite de commande, vous devez entrer

```
start cookies
```

Dans Vista, la commande doit être modifiée avec la **commande Shell** et sera donc :

```
start shell:cookies
```

De même, vous pouvez ouvrir un programme tel que Microsoft Word avec la commande

```
start winword
```

Vous pouvez également ouvrir une page Web dans Internet Explorer en utilisant une commande du type

```
start http://monsite.com
```

XXX-Q - Utilisez la complétion de nom de fichier et de nom de dossier

La fonction d'autocomplétion des noms et des chemins vous économisera de la frappe. Cette fonctionnalité utilise la touche **Tab** pour compléter automatiquement le nom de fichier que vous avez commencé à taper. Par exemple, tapez « a » (sans guillemets) dans l'invite de commande et tapez sur la touche **Tab**. Windows va alors compléter votre saisie avec le nom d'un fichier ou d'un dossier existant commençant par « a », en le sélectionnant par ordre alphabétique. Appuyez sur la touche **Tab** à nouveau pour afficher le nom de fichier ou de dossier suivant (commençant par « a »). De cette façon, vous pouvez boucler sur tous les fichiers et dossiers existant dans le chemin courant qui commence avec un caractère particulier ou un groupe de caractères. La combinaison de touches **Shift+Tab** permet de parcourir la liste dans l'ordre inverse. La fonction de complétion peut être utilisée à n'importe quel endroit d'une commande.

XXX-R - Activer le mode d'édition rapide pour la fenêtre de commande

Il est pratique d'effectuer du copier-coller à partir de la fenêtre d'invite, mais cette fonctionnalité n'est pas activée par défaut. J'utilise cette fonction fréquemment et je vous suggère de l'activer pour toutes vos fenêtres d'invite de commande. Vous trouverez plus de détail sur l'activation de l'édition rapide [sur une autre page](#). Une fois que l'édition rapide est activée, le contenu du presse-papier peut être collé dans l'invite de commande en effectuant un clic droit sur la fenêtre de commande.

XXX-S - Afficher l'historique des commandes

Le réglage par défaut de la [configuration de la fenêtre de commande](#) inclut la possibilité de mémoriser jusqu'à 50 commandes tapées. L'historique des commandes peut être affiché en appuyant sur la touche « F7 ».

XXX-T - Utiliser la commande « sleep » dans les fichiers batch sous Windows XP.

Parfois, il est pratique d'attendre un certain temps dans un fichier batch entre deux commandes. Si vous téléchargez les outils gratuits Windows Server 2003 ([décrits sur une autre page](#)), vous trouverez l'outil *sleep.exe* qui fournit un moyen d'attendre un certain temps dans un fichier batch. Pour un intervalle de *n* secondes, la commande est :

```
sleep n
```

XXX-U - Copier du texte à partir de la fenêtre de commande

À l'époque de DOS, il n'était pas rare d'entrer du texte directement de l'invite de commande dans un fichier avec la commande « copy ». C'est moins utilisé dans Windows, mais il est toujours possible de le faire. La sortie de la fenêtre de commande ou de la console est appelée CON pour « CONsole » et n'est pas sensible à la casse. Pour copier du texte depuis l'invite de commande vers un fichier « untexte.txt », la séquence d'instruction sera :

```
copy con untexte.txt
Première ligne du fichier souhaité
encore un peu de texte...
^Z
```

La dernière ligne indique la combinaison de touches « Ctrl+Z » suivie de la touche Entrée. Cette commande termine la séquence et envoie le texte au fichier qui sera automatiquement créé. Dans cet exemple particulier, le fichier est placé dans le répertoire courant, mais d'autres emplacements peuvent être utilisés.

XXX-V - Astuces pour la console de commande de Vista

Windows XP et Vista ont la plupart de leurs lignes de commande en commun. Toutefois, il existe certaines différences. Les astuces spécifiques à Windows Vista sont données au [chapitre suivant](#).

XXXI - Accéder aux répertoires systèmes avec la console de commande dans Windows Vista



Ce chapitre est une traduction de la page <http://commandwindows.com/shell.htm> dont l'auteur original est Victor Laurie.

Un certain nombre de répertoires système sont accessibles facilement dans Windows Vista en utilisant la commande « Shell: ». Les répertoires sont listés.

Vous souhaitez parfois travailler sur l'un des répertoires spéciaux du système. Comme [expliqué sur une autre page](#), les dossiers systèmes de Windows XP peuvent être ouverts en saisissant simplement leur nom dans **Démarrer-->Exécuter**. Cependant, cette procédure ne fonctionne pas dans Windows Vista. À la place, le nom de dossier doit être précédé par la commande « shell: ». Bien que l'on puisse toujours utiliser **Démarrer-Exécuter** (non affiché par défaut dans Vista), on peut aussi utiliser la zone « Rechercher » du menu Démarrer. Par exemple, pour ouvrir le répertoire « Envoyer vers », tapez

```
shell:sendto
```

Notez qu'il ne doit pas y avoir d'espace entre « shell: » et la commande. Remarquez également le symbole « : ». Cette commande n'est pas sensible à la casse.

L'invite de commande peut aussi être utilisée pour ouvrir des dossiers systèmes spéciaux avec la commande shell. Dans l'invite de commande, les commandes shell doivent être entrées en utilisant la commande « start ». Par exemple, pour ouvrir le dossier « Cookies », la commande sera :

```
start shell:cookies
```

Cette méthode pour ouvrir les dossiers systèmes fonctionne également dans Windows XP.

La liste des dossiers systèmes pouvant être accédés est donnée ci-dessous. Cette liste est extraite [de la clé du Registre](#)

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\explorer\FolderDescriptions
```

(Cette liste varie suivant la version de Vista).

XXXI-A - Dossiers systèmes de Windows Vista

AddNewProgramsFolder	MyComputerFolder
Administrative Tools	NetHood
AppData	NetworkPlacesFolder
AppUpdatesFolder	OEM Links
Cache	Original Images
CD Burning	Personal
ChangeRemoveProgramsFolder	PhotoAlbums
Common Administrative Tools	Playlists
Common AppData	PrintersFolder
Common Desktop	PrintHood
Common Documents	Profile
Common Programs	ProgramFiles
Common Start Menu	ProgramFilesX64 (in 64-bit systems)
Common Startup	ProgramFilesX86 (in 64-bit systems)
Common Templates	
CommonDownloads	

CommonMusic	ProgramFilesCommon
CommonPictures	ProgramFilesCommonX64 (in
CommonVideo	64-bit systems)
ConflictFolder	ProgramFilesCommonX86 (in
ConnectionsFolder	64-bit systems)
Contacts	Programs
ControlPanelFolder	Public
Cookies	PublicGameTasks
CredentialManager	Quick Launch
CryptoKeys	Recent
CSCFolder	RecycleBinFolder
Default Gadgets	ResourceDir
Desktop	SampleMusic
Downloads	SamplePictures
DpapiKeys	SamplePlaylists
Favorites	SampleVideos
Fonts	SavedGames
Gadgets	Searches
Games	SearchHomeFolder
GameTasks	SendTo
History	Start Menu
InternetFolder	Startup
Links	SyncCenterFolder
Local AppData	SyncResultsFolder
LocalAppDataLow	SyncSetupFolder
LocalizedResourcesDir	System
MAPIFolder	SystemCertificates
My Music	SystemX86
My Pictures	Templates
My Video	TreePropertiesFolder
	UserProfiles
	UsersFilesFolder
	Windows

XXXII - Liste des commandes de Windows Vista et Référence



Ce chapitre est une traduction de la page <http://commandwindows.com/vista-commands.htm> dont l'auteur original est Victor Laurie.

Ce chapitre contient la liste des commandes disponibles dans l'invite de commande de Windows Vista qui est similaire à celle de Windows XP, mais avec quelques fonctionnalités additionnelles. Les commandes et une brève description de leur fonction sont données ci-dessous. Quelques conseils d'utilisation sont fournis sur [une autre page](#).

XXXII-A - Commandes dans Windows Vista

À l'origine, Microsoft prévoyait d'intégrer une nouvelle invite de commande dans Vista, mais Microsoft a ensuite décidé de faire de cette nouvelle console une application indépendante. Cette nouvelle application est appelée PowerShell et est décrite sur [une autre page](#). Ainsi l'ancien interpréteur de commandes `cmd.exe` est toujours présent dans une nouvelle version. (La version de Vista est 6.0 et celle de XP est 5.1). Plusieurs commandes qui étaient auparavant disponibles uniquement dans les **Support Tools** ou dans les outils **Server 2003** sont maintenant prises en charge en standard. Par exemple, la commande `Robocopy` de Windows Server 2003 est maintenant accessible par défaut. Il y a également de nouvelles commandes, indiquées avec un astérisque dans la liste ci-dessous. Notez que certaines commandes ont besoin des privilèges administratifs. L'exécution d'une invite de commande en tant qu'administrateur est décrite [sur cette page](#). Certaines commandes spécialisées d'administration système ou réseau ont été volontairement omises.

ASSOC	Affiche ou modifie les applications associées aux extensions de fichiers.
ATTRIB	Affiche ou modifie les attributs d'un fichier.
BREAK	Active ou désactive le contrôle étendu de CTRL+C.
*BCDEDIT	Définit les propriétés dans la base de données de démarrage pour le contrôle du chargement d'amorçage.
CACLS	Affiche ou modifie les listes de contrôles d'accès aux fichiers.
CALL	Appelle un fichier de commandes à partir d'un autre fichier de commandes.
CD	Modifie le répertoire ou affiche le répertoire actif.
CHCP	Modifie ou affiche le numéro de la page de code active.
CHDIR	Modifie le répertoire ou affiche le répertoire actif.
CHKDSK	Vérifie un disque et affiche un rapport de l'état du disque.
CHKNTFS	Affiche ou modifie la vérification du disque au démarrage.
*CHOICE	Commande de fichier batch permettant à l'utilisateur de sélectionner un élément parmi une liste de choix.
CIPHER	Affiche ou modifie le cryptage des répertoires [fichier] sur les partitions NTFS.
*CLIP	Redirige la sortie d'une commande vers le presse-papier Windows.
CLS	Efface l'écran.
CMD	Exécute une nouvelle instance de l'interpréteur de commandes de Windows.
*CMDKEY	Crée, affiche et supprime les noms et mots de passe utilisateur enregistrés.
COLOR	Modifie les couleurs du premier plan et de l'arrière-plan de la console.
COMP	Compare les contenus de deux fichiers ou groupes de fichiers.
COMPACT	Modifie ou affiche la compression des fichiers sur une partition NTFS.
CONVERT	Convertit des volumes FAT en volumes NTFS. Vous ne pouvez pas convertir le lecteur en cours d'utilisation.
COPY	Copie un ou plusieurs fichiers.
DATE	Affiche ou définit la date.
DEFRAG	Utilitaire de défragmentation du disque.
DEL	Supprime un ou plusieurs fichiers.
DIR	Affiche la liste des fichiers et des sous-répertoires d'un répertoire.

DISKCOMP	Compare les contenus de deux disquettes.
DISKCOPY	Copie le contenu d'une disquette sur une autre.
DISKPART	Affiche ou configure les propriétés d'une partition de disque.
DOSKEY	Modifie les lignes de commande, rappelle des commandes Windows, et crée des macros.
DRIVERQUERY	Affiche l'état et les propriétés du pilote de périphérique en cours d'utilisation.
ECHO	Affiche des messages ou active/désactive l'affichage des commandes.
ENDLOCAL	Stoppe la localisation des modifications d'environnement dans un fichier de commandes.
ERASE	Supprime un ou plusieurs fichiers.
EXIT	Quitte l'interpréteur de commandes.
EXPAND	Décompresse un ou plusieurs fichiers compressés.
FC	Compare deux fichiers ou groupes de fichiers et affiche les différences.
FIND	Recherche une chaîne de caractères dans un ou plusieurs fichiers.
FINDSTR	Cherche des chaînes dans les fichiers.
FOR	Exécute une commande sur chaque fichier d'un ensemble de fichiers.
*FORFILES	Sélectionne un fichier (ou un ensemble de fichiers) et exécute une commande sur ce(s) fichier(s).
FORMAT	Formate un disque devant être utilisé avec Windows.
FSUTIL	Affiche ou configure les propriétés du système de fichiers.
FTYPE	Affiche ou modifie les types de fichiers utilisés dans les associations d'extensions.
GOTO	Indique l'exécution d'un fichier de commandes pour une ligne identifiée par une étiquette.
GPRESULT	Affiche les informations de stratégie de groupe pour un ordinateur ou un utilisateur.
GRAFTABL	Permet à Windows d'afficher un jeu de caractères en mode graphique.
HELP	Affiche des informations sur les commandes de Windows.
*ICACLS	Afficher, modifier, sauvegarder ou restaurer les listes de contrôle d'accès

	pour les fichiers et les répertoires (Plus de détails...).
IF	Effectue un traitement conditionnel dans un fichier de commandes.
LABEL	Crée, modifie ou supprime le nom de volume d'un disque.
MD	Crée un répertoire.
MKDIR	Crée un répertoire.
*MKLINK	Créer des liens symboliques et des liens réels.
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.
OPENFILES	Affiche les fichiers partagés ouverts à distance par les utilisateurs.
PATH	Affiche ou définit le chemin de recherche des fichiers exécutables.
PAUSE	Interrompt l'exécution d'un fichier de commandes et affiche un message.
POPD	Restaure la valeur précédente du répertoire actif enregistrée par PUSH.D.
PRINT	Imprime un fichier texte.
PROMPT	Modifie l'invite de commande de Windows.
PUSH.D	Enregistre le répertoire actif puis le modifie.
RD	Supprime un répertoire.
RECOVER	Récupère l'information lisible d'un disque défectueux.
REM	Insère des commentaires (remarques) dans un fichier batch.
REN	Renomme un ou plusieurs fichiers.
RENAME	Renomme un ou plusieurs fichiers.
REPLACE	Remplace des fichiers.
RMDIR	Supprime un répertoire.
ROBOCOPY	Utilitaire avancé pour copier les fichiers et les arborescences de répertoires.
SET	Affiche, définit ou supprime des variables d'environnement Windows.
SETLOCAL	Commence la localisation des modifications d'environnement dans un fichier de commandes.
SETX	Crée ou modifie les variables d'environnement dans l'environnement utilisateur ou système.
SC	Affiche ou configure les services (processus en arrière-plan).
SCHTASKS	Planifie les commandes et les programmes à exécuter sur l'ordinateur.
SHIFT	Modifie la position des paramètres remplaçables dans un fichier de commandes.

SHUTDOWN	Permet un arrêt local ou distant correct de l'ordinateur.
SORT	Trie les entrées.
START	Ouvre une fenêtre séparée pour l'exécution d'un programme ou d'une commande spécifique.
SUBST	Associe un chemin d'accès à une lettre de lecteur.
SYSTEMINFO	Affiche les propriétés et la configuration spécifiques de l'ordinateur.
*TAKEOWN	Cet outil permet à un administrateur de récupérer l'accès à un fichier qui avait été refusé en réassignant l'appartenance de fichier. (plus de détails...).
TASKLIST	Affiche toutes les tâches en cours d'exécution, y compris les services.
[corresp.] Taskkill	Termine ou interrompt un processus ou une application en cours d'exécution.
TIME	Affiche ou définit l'heure du système.
*TIMEOUT	Met en attente le processeur de commandes pendant le nombre de secondes spécifié. Plus de détails...
TITLE	Définit le titre de la fenêtre pour une session CMD.EXE.
TREE	Affiche graphiquement la structure des répertoires d'un lecteur ou d'un chemin d'accès.
TYPE	Affiche le contenu d'un fichier texte.
VER	Affiche la version de Windows.
VERIFY	Demande à Windows de vérifier si vos fichiers sont correctement écrits sur le disque.
VOL	Affiche le nom et le numéro de série d'un volume de disque.
*VSSADMIN	Outil en ligne de commande d'administration du service de cliché instantané de volume
*WHERE	Affiche l'emplacement des fichiers qui correspondent au modèle de recherche.
XCOPY	Copie les fichiers et les arborescences de répertoires.
WMIC	Affiche les informations WMI dans l'interface de commande interactive.

XXXIII - Astuces pour utiliser la console de commandes de Vista



Ce chapitre est une traduction de la page <http://commandwindows.com/vista-tips.htm> dont l'auteur original est Victor Laurie.

Nous allons donner ici quelques astuces spécifiques à la ligne de commande de Windows Vista.

Pour la plupart, les fonctionnalités de la ligne de commande sont les mêmes dans Windows Vista et Windows XP, mais il existe toutefois quelques différences. (La version de Vista est 6.0

XXXIII-A - Exécuter en tant qu'administrateur dans Vista

Windows Vista intègre une fonctionnalité de sécurité appelée **Contrôle de comptes utilisateurs** qui limite les privilèges des utilisateurs par défaut. Comme les lignes de commande sont souvent utilisées pour des tâches administratives, vous voudrez souvent exécuter l'invite en tant qu'administrateur. Vous pouvez faire cela en faisant un clic droit sur l'icône de l'invite de commande et en choisissant « Exécuter en tant qu'administrateur » dans le menu contextuel. Un moyen plus pratique est de mettre l'invite de commandes pour s'exécuter avec des privilèges d'administrateur par défaut. Effectuez un clic droit sur le raccourci de l'invite de commande et choisissez « Propriétés » dans le menu contextuel. Cliquez sur le bouton « Avancé » et cochez la case « Exécuter en tant qu'administrateur ». Cliquez ensuite sur « OK ». Notez que vous obtiendrez un message de confirmation quand vous ouvrirez l'invite de commande.

Un autre moyen d'accéder à l'invite de commande en tant qu'administrateur est de saisir « cmd » dans la *zone de Recherche du Menu Démarrer* et d'utiliser ensuite la combinaison clavier : **Ctrl+Maj+ Entrée**.

XXXIII-B - « Ouvrir une invite de commande ici » dans Vista

Il est désormais facile d'ouvrir une invite de commande sur un dossier de votre choix dans Vista. Si vous appuyez sur la touche **Shift** pendant que vous effectuez un clic droit sur un dossier, le menu contextuel contiendra une entrée « Ouvrir une fenêtre de commandes ici ». Si vous choisissez cet élément, une invite de commande va s'ouvrir et sera positionnée sur le dossier choisi.

XXXIII-C - Ouvrir une fenêtre de commandes avec les privilèges administrateurs à partir de n'importe où

L'option « Exécuter en tant qu'administrateur » mentionnée ci-dessus ouvre toujours l'invite sur le répertoire **Windows\System32**. Pour ouvrir une console de commandes avec les privilèges administrateur dans le répertoire de votre choix, vous pouvez ajouter une commande au menu contextuel. Le fichier INF qui effectue les changements appropriés dans le registre peut être [téléchargé ici](#). Il provient de l'utilitaire PowerToy [décrit sur cette page](#).

XXXIII-D - Envoyer la sortie d'une commande vers le presse-papier Windows avec clip.exe

Vista est fourni avec un utilitaire en ligne de commande **clip.exe** qui peut être utilisé pour rediriger la sortie d'une autre commande vers le presse-papier Windows. La commande utilise un « pipe » et est de la forme :

```
uneCommande | clip
```

Par exemple, pour envoyer un listing de répertoire vers le presse-papier, la commande est :

```
dir | clip
```


XXXIII-E - Placer le contenu d'un fichier texte dans le presse-papier Windows avec clip.exe

L'utilitaire *clip.exe* peut aussi être utilisé pour lire un fichier texte et placer son contenu dans le presse-papier de Windows. La commande à utiliser est :

```
clip < unFichier.txt
```

XXXIII-F - La commande « choice » est de retour dans Vista

Les anciens utilisateurs se rappelleront que DOS possédait une commande appelée « choice » qui permettait une interaction limitée avec l'utilisateur. La commande a ensuite été retirée des consoles de commandes 32-bits parce que la commande « set /p » fournissait une fonctionnalité similaire, voire meilleure. « choice » est de retour dans Vista sous une nouvelle forme. Entrez « choice /? » dans une invite de commandes pour avoir plus de détails sur ses fonctionnalités.

XXXIII-G - Utiliser la commande « choice » pour faire attendre un fichier batch sous Vista

Une utilisation utile de la commande « choice » est de forcer un fichier batch à se mettre en pause pendant une durée spécifiée. L'instruction est de la forme :

```
choice /T n /D y > nul
```

Le commutateur « /T n » spécifie une période d'attente de *n* secondes. Le commutateur « /D y » définit « yes » comme le choix par défaut. Pour éviter toute sortie texte non sollicitée, on redirige la sortie vers le périphérique nul.

XXXIII-H - Utiliser la commande « timeout » pour faire attendre un fichier batch dans Vista

Une autre nouvelle commande de Windows Vista est « timeout ». Celle-ci va forcer l'interpréteur de commandes à attendre un nombre de secondes spécifié ou l'appui sur une touche. Le format de la commande est :

```
timeout /T n
```

où *n* est le nombre de secondes à attendre. Pour que la commande ignore les entrées clavier, on peut utiliser le commutateur */nobreak* :

```
timeout /T n /nobreak
```

Comme la commande affiche en sortie le temps restant, il peut être utile de la redirigée vers le périphérique nul.

```
timeout /T n > nul
```

XXXIII-I - Commutateur ajouté à « Dir » pour activer l'affichage des Flux de Données Alternatifs (Alternate Data Streams)

Les fichiers NTFS **peuvent contenir des informations additionnelles** dans « streams » ou « forks ». Ces éléments ajoutés sont normalement cachés dans la plupart des fonctions de Windows comme l'Explorateur. Dans Vista, un commutateur */R* a été ajouté à la commande « dir » pour permettre l'affichage des flux de données alternatifs.

XXXIII-J - Activer le compte administrateur principal sur l'écran de connexion

Vista contient un compte administrateur principal, mais qui n'est normalement pas visible sur l'écran de connexion. Pour activer ce compte., ouvrez une fenêtre de commande avec les privilèges administrateurs et utilisez la commande

```
net user administrator /active:yes
```

(Assurez-vous de mettre un mot de passe sur ce compte). Pour enlever le compte de l'écran de connexion, utilisez la commande

```
net user administrator /active:no
```

XXXIII-K - Réduire l'espace utilisé par la Restauration Système.

La fonctionnalité de Restauration du Système peut utiliser jusqu'à 15% de l'espace disque pour le stockage de ses fichiers de sauvegarde. Comme les disques sont de plus en plus grands, cela représente un espace de plus en plus important. La commande « vssadmin » peut être utilisée pour configurer les options de la Restauration du Système. Pour contrôler l'allocation d'espace disque, ouvrez une invite de commande avec les privilèges administrateur et entrez

```
vssadmin Resize ShadowStorage /For=C: /On=C: /Maxsize=[n]
```

Remplacez [n] par la taille désirée en MB ou en GB. Les unités doivent être précisées : par exemple, « Maxsize=500MB » ou « Maxsize=2GB ». Cet exemple est pour le disque C:. **ATTENTION ! Cette opération supprime tous les anciens points de restauration !**

XXXIII-L - Augmenter le cache mémoire du système de fichier

Si vous avez tendance à ouvrir et fermer un grand nombre de fichiers, vous pouvez augmenter les performances en créant une valeur plus grande pour un réglage spécial du cache avec **l'utilitaire du système de fichier** :

```
fsutil behavior set memoryusage 2
```

D'après Microsoft, cela augmente quelque chose appelé la « paged pool » memory. N'utilisez pas ceci si vous consommez déjà de grandes quantités de mémoire système avec d'autres activités. Si les performances après ce changement ne vous satisfont pas, annulez la modification avec la commande

```
fsutil behavior set memoryusage 1
```

Ces commandes nécessitent les privilèges administrateurs. Les commandes changent un réglage dans le Registre et prennent effet après un redémarrage.

XXXIII-M - Utiliser « takeown » pour accéder à certains fichiers

Vista protège de nombreux fichiers systèmes pour des raisons de sécurité et même l'administrateur n'est pas autorisé à y accéder. Si l'accès à un fichier vous est refusé alors que vous utilisez un compte administrateur, vous pouvez utiliser l'outil en ligne de commande « takeown.exe » pour réassigner le propriétaire du fichier. Vous devez exécuter la commande depuis une invite de commande avec les privilèges administrateur. La syntaxe est :

```
takeown /f un_fichier [/a] [/r]
```

Le nom de fichier spécifié peut contenir des caractères jokers. Vous pouvez également spécifier un répertoire. Le commutateur optionnel « /a » définit le groupe des administrateurs comme nouveau propriétaire. Si ce paramètre est omis, le comportement par défaut est de définir le compte utilisateur actuellement utilisé comme nouveau propriétaire de l'élément. Le commutateur « /r » parcourt les sous-répertoires. Bien que cette commande définisse le nouveau propriétaire d'un fichier, elle ne donne pas les droits d'accès. Ainsi, si vous souhaitez modifier un fichier système (ce qui est en général une mauvaise idée), vous allez devoir appliquer la commande « icacls » décrite ci-dessous.

XXXIII-N - Obtenir les droits de contrôle sur un fichier avec « icaccls »

(icaccls.exe succède à la commande « cacls » de Windows XP. L'ancienne commande est toujours disponible). Cette commande possède un ensemble d'options complexe. Ces options peuvent être affichées en entrant « icaccls /? » dans une invite de commandes. Dans cet exemple, nous allons donner le contrôle total à un compte appelé *user* :

```
icaccls nom_de_fichier /grant user:F
```

XXXIII-O - Nettoyer les fichiers de Vista SP1

Quand vous installez le Service Pack 1 de Windows Vista, un utilitaire de désinstallation est également créé. Si vous avez installé le SP1 depuis un moment, que vous en êtes satisfait et que vous souhaitez le garder, vous pouvez supprimer les fichiers de désinstallation et libérer au moins un Go d'espace disque. Pour enlever ces fichiers de sauvegarde, utilisez la commande

```
VSP1CLN.EXE
```

J'ai utilisé les majuscules pour différencier les caractères 1 et L, mais la casse n'est pas importante). Vous devez posséder les privilèges administratifs. Après avoir exécuté cette commande, vous ne pourrez plus désinstaller le Service Pack 1 de Vista, donc soyez sûr de vouloir le garder.

XXXIII-P - Utilisation du Drag&Drop (Glisser-Déplacer)

La possibilité de glisser des commandes et de les déposer sur une fenêtre de commandes qui était disponible dans les versions antérieures de Windows ne fonctionne pas dans Vista.