

## Vue d'ensemble de l'interpréteur de commandes

L'interpréteur de commandes est un programme logiciel qui fournit une communication directe entre l'utilisateur et le système d'exploitation. L'interface utilisateur non graphique de l'interpréteur de commandes propose un environnement dans lequel vous exécutez des applications et utilitaires basés sur les caractères. L'interpréteur de commandes exécute des programmes et affiche leur sortie à l'écran en utilisant des caractères individuels, comme l'interpréteur de commandes MS-DOS Command.com. L'interpréteur de commandes de Windows XP utilise l'interpréteur de commandes Cmd.exe, qui charge les applications et dirige le flux d'informations entre les applications, afin de convertir l'entrée de l'utilisateur sous une forme que le système d'exploitation peut interpréter.

Vous pouvez utiliser l'interpréteur de commandes pour créer et éditer des fichiers de commandes (également appelés scripts) afin d'automatiser les tâches routinières. Par exemple, vous pouvez utiliser les scripts pour automatiser la gestion des comptes utilisateur ou des sauvegardes quotidiennes. En outre, l'environnement d'exécution de scripts Windows `CScript.exe` vous permet d'exécuter des scripts plus complexes dans l'interpréteur de commandes. Les fichiers de commandes vous permettent d'effectuer des opérations plus efficacement que l'interface utilisateur. Les fichiers de commandes acceptent toutes les commandes qui sont disponibles au niveau de la ligne de commande. Pour plus d'informations sur les fichiers de commandes et les scripts, consultez la section [Utilisation des fichiers de commandes](#).

Vous pouvez personnaliser la fenêtre de l'invite de commandes pour un meilleur affichage et pour mieux contrôler l'exécution des programmes. Pour plus d'informations sur la personnalisation de la fenêtre de l'invite de commandes, consultez la section [Pour configurer l'invite de commandes](#).

### Utilisation de la syntaxe de la commande

La syntaxe apparaît dans l'ordre dans lequel vous devez taper une commande et tous les paramètres qui suivent. L'exemple suivant de la commande **xcopy** présente plusieurs formats de texte de la syntaxe :

**xcopy** *Source* [*Destination*] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d:mm-jj-aaaa] [/u] [/i] [/s [/e]] [/t] [/k] [/r] [/h] [{/a/m}] [/n] [/o] [/x] [/exclude:fichier1[+fichier2][+fichier3]] [{/y/-y}] [/z]

Le tableau suivant explique comment interpréter les différents formats de texte.

### Légende de la mise en forme

Mise en forme	Signification
<i>Italique</i>	Informations que l'utilisateur doit fournir
<b>Gras</b>	Éléments que l'utilisateur doit taper tels qu'ils sont indiqués
Points de suspension (...)	Paramètre pouvant être répété plusieurs fois dans une ligne de commande
Entre crochets ([ ])	Éléments facultatifs
Entre accolades ({}), avec séparation des options par une barre verticale ( ). Exemple : {pair impair}	Ensemble d'options parmi lesquelles l'utilisateur doit en choisir une seule
Police Courier	Code ou données de programme

### Utilisation de plusieurs commandes et symboles de traitement conditionnel

Vous pouvez exécuter plusieurs commandes à partir d'une seule ligne de commande ou d'un script utilisant des symboles de traitement conditionnel. Lorsque vous exécutez plusieurs commandes avec des symboles de traitement conditionnel, les commandes à droite du symbole de traitement conditionnel agissent en fonction des résultats de la commande à gauche du symbole de traitement conditionnel. Vous pouvez par exemple exécuter une commande uniquement si la commande précédente a échoué. Vous pouvez aussi exécuter une commande uniquement si la commande précédente a réussi.

Vous pouvez utiliser les caractères spéciaux figurant dans le tableau suivant pour transmettre plusieurs commandes.

Caractère	Syntaxe	Définition
<b>&amp;</b> [...]	<i>commande1 &amp; commande2</i>	Permet de séparer plusieurs commandes sur la ligne de commande. Cmd.exe exécute la première commande, puis la seconde.
<b>&amp;&amp;</b> [...]	<i>commande1 &amp;&amp; commande2</i>	Permet d'exécuter la commande suivant <b>&amp;&amp;</b> uniquement si la commande précédant le symbole est exécutée correctement. Cmd.exe exécute la première commande, puis la deuxième commande uniquement si la première a été exécutée correctement.
		Permet d'exécuter la commande suivant <b>  </b> uniquement si la

<code>   [...]</code>	<code>commande1    commande2</code>	commande précédant <code>  </code> échoue. Cmd.exe exécute la première commande, puis la deuxième commande uniquement si la première a n'a pas été exécutée correctement (reçoit un code d'erreur supérieur à zéro).
<code>() [...]</code>	<code>(commande1 &amp; commande2)</code>	Permet de grouper ou d'imbriquer plusieurs commandes.
<code>;</code> ou <code>,</code>	<code>commande1 paramètre1;paramètre2</code>	Permet de séparer des paramètres de commande.

#### Remarques

- Le signe `&`, la barre verticale (`|`) et les parenthèses (`()`) sont des caractères spéciaux qui doivent être précédés du caractère d'échappement (`^`) ou de guillemets lorsque vous les transmettez en tant qu'arguments.
- Si une commande exécute une opération avec succès, elle renvoie un code de sortie de zéro (0) ou aucun code de sortie. Pour plus d'informations sur les codes de sortie, consultez les [Kits de ressources Microsoft Windows](#).

## Imbrication d'interpréteurs de commandes

Vous pouvez imbriquer des interpréteurs de commandes au sein de Cmd.exe en ouvrant une nouvelle instance de Cmd.exe à l'invite de commandes. Par défaut, chaque instance de Cmd.exe hérite de l'environnement de son application Cmd.exe parente. En imbriquant des instances de Cmd.exe, vous pouvez apporter des modifications à l'environnement local sans affecter l'application parente de Cmd.exe. Cette opération vous permet de préserver l'environnement d'origine de Cmd.exe et d'y retourner lorsque vous quittez l'interpréteur de commandes imbriqué. Les modifications que vous apportez dans l'interpréteur de commandes imbriqué ne sont pas enregistrées.

Pour imbriquer un interpréteur de commandes, à l'invite de commandes, tapez :

### **cmd**

Un message similaire au suivant apparaît :

```
Microsoft (R) Windows XP (TM)
(C) Copyright 1985-2001 Microsoft Corp.
```

Pour fermer l'interpréteur de commandes imbriqué, tapez **exit**.

Vous pouvez encore mieux localiser les changements dans une instance de Cmd.exe (ou dans un script) en utilisant les commandes **setlocal** et **endlocal**. La commande **Setlocal** crée une étendue locale et **endlocal** termine l'étendue locale. Toutes les modifications apportées dans l'étendue de **setlocal** et **endlocal** sont rejetées, ce qui permet de garder l'environnement initial inchangé. Vous pouvez imbriquer ces deux commandes dans 32 niveaux au maximum. Pour plus d'informations sur les commandes **setlocal** et **endlocal**, consultez [Setlocal](#) et [Endlocal](#).

## Utilisation des variables d'environnement avec Cmd.exe

L'environnement de l'interpréteur de commandes Cmd.exe est défini par des variables qui déterminent le comportement de l'interpréteur de commandes et le système d'exploitation. Vous pouvez définir le comportement de l'environnement de l'interpréteur de commandes ou l'environnement complet du système d'exploitation en utilisant deux types de variables d'environnement, système et locales. Les variables d'environnement système définissent le comportement de l'environnement global du système d'exploitation. Les variables d'environnement locales définissent le comportement de l'environnement de l'instance Cmd.exe en cours.

Les variables d'environnement système sont prédéfinies dans le système d'exploitation et sont disponibles pour tous les processus Windows XP. Seuls les utilisateurs qui disposent de privilèges d'administrateur peuvent modifier les variables système. Ces variables sont le plus souvent utilisées dans les scripts d'ouverture de session.

Les variables d'environnement locales sont disponibles uniquement lorsque l'utilisateur pour lequel elles ont été créées a ouvert une session sur l'ordinateur. Les variables locales définies dans la [ruche HKEY\\_CURRENT\\_USER](#) sont valides uniquement pour l'utilisateur en cours, mais elles définissent le comportement de l'environnement global du système d'exploitation.

La liste suivante décrit les différents types de variables, dans l'ordre de priorité décroissant :

- Variables système intégrées
- Variables système trouvées dans la ruche **HKEY\_LOCAL\_MACHINE**
- Variables locales trouvées dans la ruche **HKEY\_CURRENT\_USER**
- Tous les chemins et variables d'environnement définis dans le fichier Autoexec.bat
- Tous les chemins et les variables d'environnement définis dans un script d'ouverture de session (s'il existe)
- Variables utilisées interactivement dans un script ou un fichier de commandes

Dans l'interpréteur de commandes, chaque instance de Cmd.exe hérite de l'environnement de son application Cmd.exe parente. Par conséquent, vous pouvez modifier les variables dans le nouvel environnement Cmd.exe sans que cela n'affecte l'environnement de l'application parente.

Le tableau suivant répertorie les variables d'environnement locales et système de Windows XP.

--	--	--

Variable	Tapez	Description
%ALLUSERSPROFILE%	Locale	Renvoie l'emplacement du profil All Users (Tous les utilisateurs).
%APPDATA%	Locale	Renvoie l'emplacement auquel les application stockent les données par défaut.
%CD%	Locale	Renvoie la chaîne du répertoire en cours.
%CMDCMDLINE%	Locale	Renvoie la ligne de commande exacte qui est utilisée pour démarrer le programme Cmd.exe actuel.
%CMDEXTVERSION%	Système	Renvoie le numéro de version des extensions de l'interpréteur de commandes en cours.
%COMPUTERNAME%	Système	Renvoie le nom de l'ordinateur.
%COMSPEC%	Système	Renvoie le chemin exact de l'exécutable de l'interpréteur de commandes.
%DATE%	Système	Renvoie la date. Utilise le même format que la commande <b>date /t</b> . Générée par Cmd.exe. Pour plus d'informations sur la commande <b>date</b> , consultez <a href="#">Date</a> .
%ERRORLEVEL%	Système	Renvoie le code d'erreur de la commande la plus récente utilisée. Une valeur qui n'est pas égale à zéro indique une erreur.
%HOMEDRIVE%	Système	Renvoie la lettre du lecteur de la station de travail locale qui est connectée au répertoire de base de l'utilisateur. Définie en fonction de la valeur du répertoire de base. Le répertoire de base de l'utilisateur est spécifié dans le composant Utilisateurs et groupes locaux.
%HOMEPATH%	Système	Renvoie le chemin d'accès complet du répertoire de base de l'utilisateur. Définie en fonction de la valeur du répertoire de base. Le répertoire de base de l'utilisateur est spécifié dans le composant Utilisateurs et groupes locaux.
%HOMESHARE%	Système	Renvoie le chemin réseau du répertoire de base partagé de l'utilisateur. Définie en fonction de la valeur du répertoire de base. Le répertoire de base de l'utilisateur est spécifié dans le composant Utilisateurs et groupes locaux.
%LOGONSEVER%	Locale	Renvoie le nom du contrôleur de domaine qui a validé la session d'ouverture de session en cours.
%NUMBER_OF_PROCESSORS%	Système	Spécifie le nombre de processeurs installés sur l'ordinateur.
%OS%	Système	Renvoie le nom du système d'exploitation. Windows 2000 affiche le système d'exploitation Windows_NT.
%PATH%	Système	Spécifie le chemin pour la recherche des fichiers exécutables.
%PATHEXT%	Système	Renvoie la liste des extensions de fichier que le système d'exploitation considère comme étant exécutables.
%PROCESSOR_ARCHITECTURE%	Système	Renvoie l'architecture du processeur. Les valeurs sont : x86, IA64.
%PROCESSOR_IDENTIFIER%	Système	Renvoie la description du processeur.
%PROCESSOR_LEVEL%	Système	Renvoie le numéro du modèle de processeur installé sur l'ordinateur.
%PROCESSOR_REVISION%	Système	Renvoie le numéro de révision du processeur.
%PROMPT%	Locale	Renvoie les paramètres de l'invite de commandes de l'interpréteur en cours. Générée par Cmd.exe.
%RANDOM%	Système	Renvoie un nombre décimal compris entre 0 et 32767. Générée par Cmd.exe.
%SYSTEMDRIVE%	Système	Renvoie le lecteur contenant le répertoire racine de Windows XP (c'est-à-dire la racine système).
%SYSTEMROOT%	Système	Renvoie l'emplacement du répertoire racine de Windows XP.
%TEMP% et %TMP%	Système et utilisateur	Renvoie les répertoires par défaut temporaires utilisés par les applications disponibles pour les utilisateurs qui ont ouvert une session. Certaines applications requièrent TEMP et d'autres TMP.
%TIME%	Système	Renvoie l'heure. Utilise le même format que la commande <b>time /t</b> . Générée par Cmd.exe. Pour plus d'informations sur la commande <b>time</b> , consultez <a href="#">Time</a> .
%USERDOMAIN%	Locale	Renvoie le nom du domaine qui contient le compte de l'utilisateur.
%USERNAME%	Locale	Renvoie le nom de l'utilisateur qui a ouvert une session.
%USERPROFILE%	Locale	Renvoie l'emplacement du profil de l'utilisateur en cours.
%WINDIR%	Système	Renvoie l'emplacement du répertoire du système d'exploitation.

## Définition des variables d'environnement

Utilisez les commandes **set** pour créer, modifier, supprimer ou afficher des variables d'environnement. La

commande **set** modifie les variables uniquement dans l'environnement de l'interpréteur de commandes en cours.

Pour visualiser une variable, à l'invite de commandes, tapez :

**set** *NomVariable*

Pour ajouter une variable, à l'invite de commandes, tapez :

**set variablename=***valeur*

Pour supprimer une variable, à l'invite de commandes, tapez :

**set** *NomVariable*=

Vous pouvez utiliser la plupart des caractères en tant que valeurs de variables, y compris les espaces. Si vous utilisez les caractères spéciaux <, >, |, &, ou ^, vous devez les faire précéder d'un caractère d'échappement (^) ou de guillemets. Si vous utilisez les guillemets, ils sont intégrés dans la valeur car tous les éléments qui suivent le signe égal sont considérés comme la valeur. Considérons les exemples suivants :

- Pour créer la valeur de la variable **new&name**, tapez :  
**set varname=new^&name**
- Pour créer la valeur de la variable **"new&name"**, tapez :  
**set varname="new&name"**
- Si vous tapez **set varname=new&name** à l'invite de commandes, un message d'erreur semblable au suivant apparaît :

```
"name" n'est pas reconnu comme commande interne ou externe, programme ou fichier de commandes."
```

Les noms de variables ne respectent pas la casse. Cependant, la commande **set** affiche la variable telle que vous l'avez tapée. Vous pouvez associer des lettres en majuscules et des lettres en minuscules dans les noms de variables pour que le code soit plus lisible (par exemple NomUtilisateur).

#### Remarques

- La taille maximale d'une variable d'environnement est 8 192 octets.
- La taille maximale des variables d'environnement totales pour toutes les variables, qui inclut le nom des variables et le signe égal, est 65 536 Ko.

## Remplacement des valeurs de variables d'environnement

Pour autoriser le remplacement des valeurs de variables dans la ligne de commandes ou dans des scripts, insérez le nom de la variable entre les signes de pourcentage (**%nomvariable%**). Lorsque vous utilisez les signes de pourcentage, vous vous assurez que Cmd.exe référence les valeurs de variables au lieu d'effectuer une comparaison littérale. Une fois que vous avez défini les valeurs de variables d'un nom de variable, insérez le nom de la variable entre les signes de pourcentage. Cmd.exe effectue une recherche sur toutes les instances du nom de la variable et le remplace par la valeur de variable définie. Par exemple, si vous créez un script qui contient différentes valeurs (par exemple, noms utilisateur) et si vous voulez définir la variable d'environnement USERNAME de chaque utilisateur avec ces valeurs, vous pouvez écrire un script en utilisant la variable USERNAME insérée entre les signes de pourcentage. Lorsque vous exécutez ce script, Cmd.exe remplace %USERNAME% par les valeurs de variables. Par conséquent, il n'est plus nécessaire d'effectuer cette tâche manuellement pour chaque utilisateur. Le remplacement de variables n'est pas récursif. Cmd.exe vérifie une seule fois les variables. Pour plus d'informations sur le remplacement de variables, consultez [For](#) et [Call](#).