

## Complément du manuel

<i>Nouvelles commandes</i> .....	1
ExecuteAddon .....	1
FileExists .....	1
PrintSetup .....	1
SendKeys .....	2
<i>Commandes modifiées</i> .....	2
DOSCommand .....	2
FileRead .....	3
FileWrite .....	3
SystemInfo .....	3
<i>Variables</i> .....	4
Variables composées et tableaux .....	4
Variables globales .....	4
<i>Nouvelles options de paramétrage du projet</i> .....	5
Contrôle à partir du clavier : .....	5
Economiseur d'écran : .....	6
Divers : .....	6
<i>Modifications du compilateur de NeoBook</i> .....	6

## Nouvelles commandes

Les commandes suivantes ont été ajoutées à la version 3.2 de NeoBook :

### **ExecuteAddon**

Exécuter un module externe. Les commandes réclamées sous forme de paramètre dépendent du module utilisé. Veuillez vous reporter à la documentation du module externe pour obtenir davantage de précisions.

#### **Syntaxe :**

```
ExecuteAddon "nom du module externe" "commandes"
```

#### **Exemple :**

```
ExecuteAddon "MonModule.exe" "MaFonction"
```

### **FileExists**

Déterminer si un fichier externe existe. Si le fichier est trouvé, la variable spécifiée comme second paramètre contiendra la valeur "True" sinon elle contiendra "False".

#### **Syntaxe :**

```
FileExists "nom de fichier" "[nom de variable]"
```

#### **Exemple :**

```
FileExists "c:\autoexec.bat" "[Résultat]"  
If "[Résultat]" "=" "True"  
    AlertBox "Etat" "Le fichier existe !"  
EndIf
```

### **PrintSetup**

Afficher la boîte de dialogue *Configuration de l'impression* (standard de Windows) et stocker la réponse de l'utilisateur ("True" = OK pour imprimer, "False" = Annuler l'impression) dans une variable. Normalement, les commandes de NeoBook relatives à l'impression (PrintPage, PrintTextFile, etc.) affichent automatiquement la boîte de dialogue *Configuration de l'impression* et demandent à l'utilisateur

une confirmation avant d'imprimer. Il existe des situations dans lesquelles l'auteur d'une présentation peut souhaiter contrôler le moment où apparaît la boîte de dialogue *Configuration de l'impression*. C'est pourquoi NeoBook propose la commande PrintSetup ainsi qu'une option dans le paramétrage du projet (voyez ci-après la rubrique Nouvelles options de paramétrage du projet) pour désactiver l'affichage automatique de la boîte de dialogue *Configuration de l'impression*.

**Syntaxe :**

```
PrintSetup "[nom de variable]"
```

**Exemple :**

```
PrintSetup "[Résultat]"
If "[Résultat]" "=" "True"
    PrintPage "Sommaire"
    PrintPage "Aide"
EndIf
```

## SendKeys

Envoyer des séquences de touches à une autre application Windows en cours d'exécution. Cette commande vous permet par conséquent de piloter une autre application depuis NeoBook. Si le programme n'est pas chargé, vous devez au préalable utiliser la commande DOSCommand pour l'activer avant de pouvoir lui envoyer des séquences de touches. Celles-ci peuvent comporter du texte, ainsi que des touches de fonction dont la liste figure ci-après :

{BS}	{F2}	{F7}	{F12}	{Left}	{Del}
{Tab}	{F3}	{F8}	{Home}	{Right}	
{Enter}	{F4}	{F9}	{End}	{PgUp}	
{Esc}	{F5}	{F10}	{Up}	{PgDn}	
{F1}	{F6}	{F11}	{Down}	{Ins}	

**Note :**

BS = Espace arrière  
Enter = Entrée  
Left = Gauche  
Right = Droite  
Up = Haut  
Down = Bas  
Home = Origine  
Del = Suppr  
End = Fin  
PgUp = Page précédente  
PgDn = Page suivante

Vous pouvez également spécifier l'état (down [=pressé] ou up [=non pressé]) des touches Shift (=Maj), Control et Alt afin de pouvoir utiliser la totalité des combinaisons de touches.

Par exemple : Alt+A s'écrira {AltDn}A{AltUp}.

Ces touches doivent toujours être utilisées par paire.

Par exemple, si vous utilisez {ShiftDn}, vous devez penser à utiliser un code {ShiftUp}, sinon votre clavier va se comporter comme si la touche Shift (=Maj) était tout le temps enfoncée.

Exemples de paires :

{ShiftDn}	{ShiftUp}
{CtrlDn}	{CtrlUp}
{AltDn}	{AltUp}

**Syntaxe :**

```
SendKeys "nom du programme" "séquence de touches"
```

**Exemple :**

```
SendKeys "notepad.exe" "Bonjour, tout le monde{Enter} {AltDn}FP{AltUp}"
```

## Commandes modifiées

Ces commandes, déjà présentes dans NeoBook, sont améliorées dans la version 3.2.

### DOSCommand

Cette commande permet d'exécuter une application, un programme ou une commande externe DOS ou Windows.

Le premier paramètre (commande) indique le nom et le chemin du programme devant être lancé.

Le second paramètre (paramètres) indique les items devant être passés au programme lorsqu'il est lancé (un nom de fichier pour un document à charger, des options pour une ligne de commande, etc.)

Le troisième paramètre (options), nouveau dans la version 3.2, peut être soit "Normal" pour lancer le programme normalement ou "Wait" pour suspendre le fonctionnement de la publication NeoBook jusqu'à la fin de l'exécution du programme lancé. Vous pouvez, de façon optionnelle, combiner les deux possibilités "Normal" et "Wait" avec "RunOnce" pour éviter que le programme soit lancé plusieurs fois simultanément.

**Syntaxe :**

```
DOSCommand "commande" "paramètres" "options"
```

**Exemple :**

```
DOSCommand "c:\windows\notepad.exe" "c:\autoexec.bat" "Normal+RunOnce"
```

### **FileRead**

Lire des données se trouvant dans un fichier externe, à une ligne dont le numéro est spécifié.

Les données que contient la ligne seront stockées dans la variable spécifiée. Ce qui est nouveau dans la version 3.2, c'est la directive spéciale "All" qui peut être utilisée à la place d'un numéro de ligne afin de lire l'intégralité du fichier. L'exemple suivant utilise la commande FileRead pour construire un menu à partir du contenu du fichier de données PARTS.DAT. Chaque ligne est lue et placée dans la variable [ChoixMenu] et les options du menu sont affichées à l'aide de la commande MenuEx.

**Syntaxe :**

```
FileRead "nom du fichier" "numéro de la ligne" "[nom de variable]"
```

**Exemple :**

```
FileRead "parts.dat" "All" "[ChoixMenu]"  
MenuEx "-1" "-1" "[ChoixMenu]" "[Résultat]"
```

### **FileWrite**

Ecrire des données dans un fichier externe, à une ligne dont le numéro est spécifié. Utiliser la directive spéciale "Append" à la place du numéro de la ligne pour ajouter les données à la fin du fichier. Vous pouvez aussi utiliser la nouvelle directive "All" pour remplacer la totalité du fichier.

L'exemple ci-après permet de remplacer la totalité du fichier par le contenu de la variable [Parts].

**Syntaxe :**

```
FileWrite "nom du fichier" "numéro de la ligne" "données"
```

**Exemple :**

```
FileWrite "parts.dat" "All" "[Parts]"
```

### **SystemInfo**

Obtenir des informations concernant cet ordinateur et stocker le résultat dans une variable. Les types d'informations sont les suivants : PubAuthor, PubColors, PubDir, PubHeight, PubWidth, PubTitle, ScreenColors, ScreenHeight, ScreenWidth, SystemDir, TempDir et WindowsDir.

La version 3.2 de NeoBook ajoute : CDRomDrive, CommandLine, NetworkDrive, WindowsVer, WindowsPlatform, WinHandle et UserName.

(Vous pouvez vous reporter à la section Variables globales pour obtenir un descriptif plus précis des nouvelles informations)

**Note :** *A partir de la version 3.2 de NeoBook, la commande SystemInfo devient obsolète dans la mesure où les items mentionnés ci-dessus peuvent simplement être référencés comme des variables globales au sein de la présentation. (Voyez la section Variables globales ci-après) La commande SystemInfo est toujours disponible afin de garantir la compatibilité avec les publications créées avec des versions précédentes de NeoBook.*

Dans l'exemple ci-après, le mode graphique de l'ordinateur de l'utilisateur est comparé au mode graphique utilisé dans la publication, et un message est affiché si l'ordinateur de l'utilisateur utilise un mode offrant une résolution moindre :

**Syntaxe :**

```
SystemInfo "type d'informations" "[variable name]"
```

**Exemple :**

```
SystemInfo "ScreenColors" "[ModeEcran]"  
SystemInfo "PubColors" "[ModePublication]"  
If "[ModePublication]" ">" "[ModeEcran]"  
    AlertBox "Alerte" "Ce programme demande un mode [ModePublication]couleurs pour|être  
    affiché correctement. |Du fait que Windows est actuellement en mode [ModeEcran]  
    couleurs|vous ne pourrez pas voir ce programme dans les meilleures conditions."  
EndIf
```

Dans NeoBook 3.2, vous pouvez simplifier ce script en faisant référence directement aux variables globales ScreenColors et PubColors :

```
If "[PubColors]" ">" "[ScreenColors]"
```

```
AlertBox "Notice!" "Ce programme demande un mode [PubColors] couleurs pour |être affiché  
correctement. |Du fait que Windows est actuellement en mode [ScreenColors] couleurs, |vous  
ne pourrez pas voir ce programme dans les meilleures conditions."
```

```
EndIf
```

## Variables

NeoBook vous permet d'utiliser ce qu'il appelle une variable au sein des paramètres d'une commande ou dans un texte de titre en complément ou en remplacement d'un texte. Certaines commandes avancées utilisent même des variables pour vous restituer des informations. Une variable est une zone de la mémoire de l'ordinateur que vous pouvez utiliser pour stocker temporairement des informations (texte ou nombre) concernant votre présentation. Par exemple, vous pouvez demander à l'utilisateur de la présentation d'entrer son nom au lancement de celle-ci. Par la suite, cette variable peut être employée pour enregistrer le résultat d'un test ou pour personnaliser la présentation en y intégrant le nom de l'utilisateur. Chaque variable utilisée dans une présentation se voit attribuer un nom unique, et le choix est pratiquement sans limite. Dans NeoBook, les noms de variables doivent obligatoirement être entourés de [crochets]. C'est le moyen pour NeoBook de savoir par exemple si vous parlez de la variable [test] ou du mot "test".

Voici quelques exemples de noms de variables :

```
[Réponse] [Nom] [Prix] [Score] [X]
```

Faire référence à une variable comme paramètre dans une commande est vraiment très simple :

```
AlertBox "Quelques civilités" "Bonjour [Nom]."
```

### Variables composées et tableaux

NeoBook 3.2 vous permet d'utiliser des noms de variables composées pour réaliser des tableaux de variables. Par exemple, une variable composée telle que [Noms[X]], dans laquelle [X] est une valeur numérique pouvant être incrémentée, peut être utilisée pour charger un tableau de variables à partir d'un fichier :

```
SetVar "[Nom du fichier]" "c:\MesFichiers\noms.dat"  
FileLen "[Nom du fichier]" "[Longueur]"  
SetVar "[X]" "1"  
While "[X]" "<=" "[Longueur]"  
    FileRead "[Nom du fichier]" "[X]" "[Noms[X]]"  
    SetVar "[X]+1" "[X]"  
EndWhile
```

Les items individuels dans le tableau Noms sont accessibles de la façon suivante :

```
[Noms1] [Noms2] etc.
```

### Variables globales

NeoBook inclut plusieurs variables globales prédéfinies. Contrairement aux variables que vous créez, celles-ci sont automatiquement prise en charge et mises à jour par NeoBook. La plupart des variables globales peuvent être insérées aux mêmes endroits que les variables normales. Certaines de ces variables ne sont accessibles qu'en lecture, ce qui signifie que leur valeur est fixe et ne peut pas être modifiée. Voici les variables globales et en lecture seule disponibles dans NeoBook 3.2:

[CDRomDrive]	lettre du premier lecteur de CD-ROM ou "erreur" en cas d'absence
[CommandLine]	paramètres de la ligne de commande transmis à la publication
[DateLong]	la date au format long de Windows (par exemple, 22 octobre 1998)
[DateShort]	la date au format court de Windows (par exemple, 22/10/98)
[Day]	le jour sous forme de texte (par exemple, Lundi)
[DayNum]	la date actuelle
[Hour]	l'heure actuelle
[Minute]	la minute actuelle
[Month]	le mois sous forme de texte (par exemple, Octobre)
[MonthNum]	le mois sous forme de nombre (par exemple, 10)
[NetworkDrive]	lettre du premier lecteur du réseau ou "erreur" en cas d'absence
[P]	idem [PageNumber]
[PageNumber]	le numéro de la page active
[PageTitle]	le titre de la page active

[PubAuthor]	l'auteur de la publication comme indiqué dans les paramètres du projet
[PubColors]	le nombre maxi de couleurs de la publication comme indiqué dans les paramètres du projet
[PubDir]	le lecteur et le répertoire dans lequel se trouve la publication exe
[ScreenColors]	le nombre maxi de couleurs autorisé par la configuration vidéo de Windows
[ScreenHeight]	la hauteur en pixels d'un écran Windows complet
[ScreenWidth]	la largeur en pixels d'un écran Windows complet
[Second]	la seconde actuelle
[SystemDir]	l'emplacement du répertoire System de Windows (en général c:\windows\system)
[TempDir]	l'emplacement du répertoire temporaire de Windows
[Time]	l'heure (H:M:S AM/PM)
[Time24]	l'heure actuelle au format 24 heures
[UserName]	le nom de l'utilisateur actuellement connecté (si un réseau est installé)
[WindowsDir]	le répertoire dans lequel Windows est installé (en général c:\windows)
[WindowsVer]	le numéro de version de Windows (unité + décimale, exemple : 4.0 pour Windows 95)
[WindowsPlatform]	la plate-forme Windows actuellement installée : Windows 95 = 1, Windows NT = 2)
[WinHandle]	le pointeur sur la publication elle-même (réservé pour une utilisation ultérieure par des modules externes)
[Year]	l'année en cours (par exemple, 1998)

Venant en complément des variables globales et en lecture seule, il existe des variables qui peuvent être lues mais dans lesquelles il est également possible d'écrire :

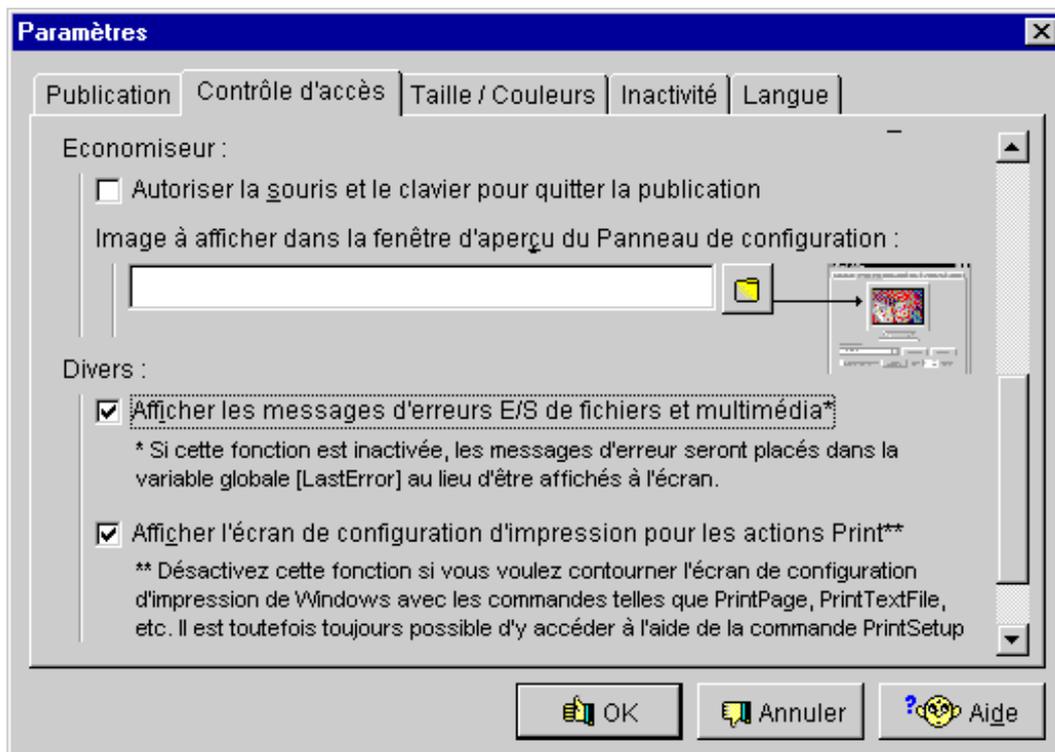
[Clipboard]	le contenu du presse-papiers de Windows (au format texte seulement). Par exemple, pour placer du texte dans le presse-papiers, vous pouvez utiliser la commande SetVar : SetVar "[Clipboard]" "placer ceci dans le presse-papiers"
[PubWidth]	la largeur de la partie intérieure de la fenêtre de la publication (l'espace de travail)
[PubHeight]	la hauteur de la partie intérieure de la fenêtre de la publication (l'espace de travail)
[PubTitle]	le contenu de la barre de titre de la fenêtre de la publication
[WindowLeft]	l'abscisse du coin supérieur gauche de la fenêtre de la publication
[WindowTop]	l'ordonnée du coin supérieur gauche de la fenêtre de la publication
[WindowWidth]	la largeur de la fenêtre de la publication y compris le cadre et les ascenseurs (éventuels)
[WindowHeight]	la hauteur de la fenêtre de la publication y compris la barre de titre, le cadre et les ascenseurs (éventuels)
[WindowState]	l'état de l'affichage de la fenêtre de la publication (Normal, Réduit ou Agrandi). Vous pouvez changer cet état en modifiant le contenu de la variable. Par exemple : SetVar "[WindowState]" "Minimized"

## Nouvelles options de paramétrage du projet

Plusieurs nouvelles options ont été ajoutées à la page Contrôle d'accès de la boîte de dialogue *Paramètres*.

### *Contrôle à partir du clavier :*

La nouvelle option **Afficher des pointillés visibles pour indiquer l'item actif** est conçue pour être utilisée en conjonction avec les options Autoriser Tab, Maj+Tab et Entrée pour activer les boutons ou le défilement (texte et images). Utilisez cette fonctionnalité pour désactiver l'affichage du rectangle en pointillé qui entoure l'item actif lors du test ou de l'exécution d'une publication. Cette fonctionnalité n'est accessible que si l'une des options Autoriser Tab... est activée.



### **Economiseur d'écran :**

Ces fonctionnalités sont utiles si vous avez l'intention de compiler votre publication sous forme d'économiseur d'écran. Cochez la case Autoriser la souris et le clavier pour quitter la publication pour que l'économiseur se désactive lorsque l'utilisateur déplace la souris ou presse sur une touche - exactement comme le font les économiseurs classiques. Si cette fonction est désactivée, votre économiseur d'écran ne se désactive que lorsque la touche Echap est pressée. L'image à afficher dans la fenêtre d'aperçu du Panneau de configuration est une petite image BMP statique qui apparaît lors de la sélection parmi les économiseurs d'écran installés.

### **Divers :**

Le fait de désactiver l'option **Afficher les messages d'erreurs E/S de fichiers et multimédia** permet à des auteurs NeoBook confirmés de piéger certains messages d'erreur et d'y réagir par programmation plutôt que de laisser NeoBook afficher l'erreur dans une boîte de dialogue. Lorsque l'option est désactivée, les messages d'erreur sont placés dans la variable [LastError]. Vous pouvez alors l'utiliser dans vos scripts pour déterminer le moment où se produit l'erreur et la façon d'y répondre. Par exemple :

```
FileWrite "parts.dat" "Append" "[PartNum],[Prix]"
If "[LastError]" ">" ""
    MessageBox "Erreur" "[LastError]"
EndIf
```

Le fait de désactiver l'option **Afficher l'écran de configuration d'impression pour les actions Print** empêche l'affichage de la boîte de dialogue **Configuration de l'impression** (standard de Windows), et permet à des commandes faisant appel à l'imprimante (PrintPage, PrintTextFile, etc.) de lancer l'impression sans confirmation. Lorsque cette fonction est désactivée, il est possible d'afficher la boîte de dialogue **Configuration de l'impression** par l'intermédiaire de la commande PrintSetup. Par exemple :

```
PrintSetup "[Résultat]"
If "[Résultat]" "=" "True"
    PrintPage "Sommaire"
    PrintPage "Aide"
EndIf
```

## **Modifications du compilateur de NeoBook**

Le compilateur de NeoBook inclut désormais une fonction de compression plus puissante qui ne se limite pas à la réduction du poids des images de la publication, mais qui compresse les polices de caractères, les séquences vidéo et les fichiers sons. Avec l'apparition de la version 3.2, l'option **Compresser les images** s'est transformée en **Compresser la publication** pour mieux faire ressortir cette nouvelle possibilité. En fonction du contenu de la présentation, cette fonction peut permettre de réduire considérablement la taille des publications compilées.